UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LOCK-AND-KEY SECURITY

EVALUATION OF TELNET AS AN AUTHENTICATION METHOD USUALLY ASSOCIATED WITH DYNAMIC ACCESS CONTROL LISTS APPLICATION

THESIS

PRESENTED

AS A PARTIAL REQUIREMENT FOR

THE MASTER PROGRAM OF MANAGAMENT INFORMATION SYSTEMS

BY

RIHAM ELSAADANY

MAY 2013

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SÉCURITÉ DE SERRURE ET CLÉ

ÉVALUATION DU TELNET COMME ÉTANT UNE MÉTHODE D'AUTHENTIFICATION SOUVENT ASSOCIÉE À L'APPLICATION DES LISTES DE CONTRÔLE D'ACCÈS DYNAMIQUES

MÉMOIRE

PRESENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAITRÎSE EN INFORMATIQUE DE GESTION

PAR

RIHAM ELSAADANY

MAI 2013

ACKNOWLEDGMENTS

I'd like to thank Mr. Guy Begin, my research director who helped me with every reference, encouraged me when it seemed endless and guided me during my journey writing this research while being in a different country. I really appreciate all his great help and generous support.

I would like to thank Mr. Cloutier, the director of the Management Information Systems master program, who made this work possible to happen, allowing me to continue my studies and giving me the chance to earn my degree after many years away from school.

My thanks also go for Mrs. Côté, the secretary of the Management Information Systems master program, who facilitated and coordinated a lot of the paper work for me. Without her great help, such a research would've been impossible.

And last, but definitely not least, I would like to thank my loving husband that supported me all the way, in every mean, so that I can achieve my goal, no matter how hard some days used to be. And, of course, many cute thanks for my three kids that tolerated a mom with less time to cuddle, however with a loving heart and mind, that never stop thinking about them... all the time.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF TERMINOLOGIES/ACRONYMS

AAA	Authentication, Authorization and Accounting
ACLs	Access Control lists
ARA	AppleTalk Remote Access
ARAP	Appletalk Remote Access Protocol
AS	Authentication Server
CHAP	Challenge Handshake Authentication Protocol
CLI	Command Line Interface
DNS SRV	Domain Name Server Service
IP	Internet Protocol
MD5	Message Digest algorithm for generating hash passwords
NAPTR	Name Authority Pointer
NAS	Network Access Server
NASI	Novel Asynchronous Service interface
OSI model	Open System Interconnection network architecture model
OTP	One Time Password
PAP	Password Authentication Protocol
PPP	Point-to-Point connection
RADIUS	Remote Authentication Dial In User Server
RAM	Random Access Memory
RFC	Request for Comments
RSA	Rivest, Shamir, Adleman public Key cryptography algorithm

SLIP	Serial Line Internet Protocol
SMS	Short Message Service
SSE	Silicon Switching Engine
SSH	Secure Shell
TACACS+	Terminal Access Control Access Control System Plus
TCP	Transport Control Protocol
TCP/IP model	Transport Control Protocol/Internet Protocol network architecture model
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TTY	Teletype printer
UDP	User Datagram Protocol
VPN	Virtual Private Networks
VSA	Vendor Specific Attribute
VTY	Virtual Terminal Line

WRITING AND TECHNICAL CONVENTIONS

- The term "He" will be used within the study, in order to refer to the human being regardless of his gender (male or female).
- The network topologies stated within this study are assumed to refer to operating networks with no connectivity issues throughout all the layers of the OSI model architecture; since connectivity probabilities and troubleshooting is out of the scope of this study.
- All internetworked devices described in the document are assumed to be Cisco devices (Juniper and other vendor devices are out of the scope of this study), even though the interoperability subject related to different authentication methods/devices will be covered by the study.
- The Cisco Internetwork Operating System (IOS release 12.3(1) and up) is assumed to be well installed and operating upon all the internetworked devices described in the study including routers and switches. Thus all the configuration codes stated within the study will only work well with Cisco IOS, excluding other vendors' operating systems.
- All internetworked devices described in the document are assumed to be well configured according to best practice guidelines, as well as according to the topology interconnectivity needs (default configuration necessary adjustments are out of the scope of this study).
- The term "user" is used to refer to the person who uses a service or a program.
- The term "client" is used to refer to the device or the program needing a service.
- The term "server" is used to refer to the program or the device providing a service for a client.
- The term "Service" is used to refer to some actions to be performed by a server program.
- The term "credentials" is used to refer to the user's login information used to authenticate the user.
- The term "topology" is used to refer to a given network architectural design.

COMMAND SYNTAX CONVENTIONS

- **Bold** indicates commands and keywords to be literally entered in the configuration.
- *Italic* indicates arguments or variables that should be substituted by an actual value.
- Vertical bars (|) separate alternative exclusive elements.
- Square brackets ([ ]) indicate options.
- Braces ({ }) indicate a required choice to be made out of a list of elements.
- Braces within brackets ([{ }]) indicate a required choice to be made out of a list of elements, within an optional element.

RÉSUMÉ

Dans les systèmes sans réseaux de communications ou bien les organisations juste avec intranet, les différentes machines et ressources sont souvent totalement isolées, ou bien accédées juste via l'intranet de l'entreprise; elles sont donc utilisées par les usagers éprouvés et autorisés par l'organisation. Les ressources de l'organisation de nos jours qui sont en réseau, tout en étant interconnectées par l'Internet, sont autrement toujours sujettes aux attaques réseautiques venant d'un nombre illimité d'usagers. Ainsi, dans les environnements informatiques en temps partagé, le système d'exploitation, aussi que d'autres mécanismes de sécurité, protègent les ressources bien que les usagers l'un de l'autre. Une telle protection de sécurité prend lieu par l'établissement de quelques règles d'accès pour les différents types d'utilisateurs. Afin de classifier les utilisateurs et les faire correspondre à leurs règles d'accès selon leurs droits d'accès, l'utilisateur doit s'identifier au processus de sécurité dès sa connexion aux ressources de l'organisation, ce processus est appelé: *l'authentification de l'utilisateur*. L'authentification de l'utilisateur est une pierre angulaire pour la sécurité réseautique de toutes les organisations, ainsi, c'est un des sujets principaux analysés par cette étude. Ce sujet va être élaboré en plus de détails dans chapitre IV, titré *L'authentification de l'utilisateur.*

Comme le besoin des utilisateurs pour accéder via le Web aux ressources internes[1] de différentes organisations a dernièrement émergé (afin d'accéder aux serveurs internes de leurs lieux de travail ou bien ceux des organisations offrant des différents services web), le besoin d'autoriser les utilisateurs a aussi augmenté (afin de sécuriser un tel accès à distance). Cela va nous amener à analyser l'usage des listes de contrôle d'accès dynamiques. Les listes de contrôle d'accès dynamiques sont des essentiels moyens de sécurité qui permettent à l'utilisateur d'accéder en sécurité aux ressources internes d'une organisation, tout en connectant à distance. Les listes de contrôle d'accès dynamiques dépendent complètement de l'authentification de l'utilisateur comme étant une garantie de sécurité de l'identité de l'utilisateur. Les listes de contrôle d'accès dynamiques vont être décrites en détail dans chapitre II, titré *Les listes de contrôle d'accès.*

D'après Odom (Odom, 2009), Telnet est le protocole d'émulation de terminal standard de la couche d'application dans le TCP/IP empilage. Telnet est utilisé pour la connexion à distance au terminal, permettant les utilisateurs d'accéder aux systèmes distants et d'utiliser les ressources comme si elles étaient connectées à un système local. Telnet est défini dans RFC 854 et sera analysé en détail dans chapitre III comme étant une méthode d'authentification utilisée dans la configuration des listes de contrôle d'accès dynamiques.

Les listes de contrôle d'accès dynamiques utilisent souvent Telnet comme une méthode d'authentification des utilisateurs. Cependant, Telnet est caractérisé par un nombre

[1] Les ressources internes sont situées sur le réseau d'une organisation. Ce réseau est supposé d'être solidement protégé contre les menaces de sécurité, incluant les accès non-autorisés.

de désavantages de sécurité, ce qui ne peut pas garantir une authentification d'utilisateurs complément sécurisée. Ainsi, l'utilisation de Telnet pour établir le processus de l'authentification de l'utilisateur des listes de contrôle d'accès dynamiques est toujours sujette à exposer les ressources internes de l'organisation à plusieurs menaces et brèches de sécurité. À cause de ces raisons, notre étude a eu lieu afin d'évaluer Telnet comme étant une méthode d'authentification, à élaborer ses avantages et ses inconvénients et à suggérer des méthodes d'authentification alternatives qui peuvent être utilisées dans la configuration des listes de contrôle d'accès dynamiques, afin d'authentifier les utilisateurs à distance. Ainsi, dans chapitre 5, Telnet sera analysé comme étant une méthode d'authentification comparée à d'autres méthodes d'authentification utilisées dans la configuration des listes de contrôle d'accès dynamiques, comme les serveurs d'authentification, incluant TACACS+, RADIUS, DIAMETER et Kerberos.

L'étude inclut un nombre important de codes de configuration qui sont spécifiquement développés afin d'appuyer les concepts de sécurité analysés et afin de présenter des directives pour guider les concepteurs des réseaux à faire de bons choix de sécurité, garantissant aux utilisateurs une connexion à distance plus sécurisée.

Mots-clés:

AAA server comparison, Access Control Lists, Authentication method lists, Authentication server, DIAMETER, Dynamic ACLs Authentication, KERBEROS, Network architecture, Network Security, OSI model, Proxy servers, RADIUS, SSH, Standard ACLs, TACACS+, TCP/IP model, Telnet, VPN.

ABSTRACT

In non-network systems or in intranet organizations, different machines and resources are either totally isolated, or only reached through the enterprise Intranet, therefore they're used by the allowed trusted users of the organization. Interconnected organization resources, which are nowadays resources connected through the Internet, on the other hand, are always prone to network attacks from an unlimited number of users. Thus, in time-sharing computing environment, the operating system, as well as other security mechanisms, protect resources and users from one another. Such security protection takes place by setting some access rules to different kinds of users. In order to classify the users and to match them to their right of access rules, the user has to identify himself to the security process once the he logs into the organization resources, this process is called: the user authentication.

User Authentication is a cornerstone in any organization's network security; thus it is one of our main subjects analyzed in this study. This subject will be elaborated in more details in Chapter 4, titled User Authentication.

As the need lately emerged for the users to access the inner resources[2] of different organizations through the web (either to access the inner servers of their work places or those of other organizations providing services), the need for user authorization also rose (in order to secure such a remote access). This will bring us to analyze the use of Dynamic Access Control Lists (ACLs). Dynamic ACLs are essential security means that allow a user to securely access an organization's inner resources while being remotely logged in. Dynamic ACLs depend completely on the user authentication as a security guarantee of the identity of the user. Dynamic ACLs will be described in details in Chapter 2, titled Access Control Lists.

According to Odom (Odom, 2009), Telnet is the standard terminal-emulation application layer protocol in the TCP/IP protocol stack. Telnet is used for remote terminal connection, enabling users to log in to remote systems and use resources as if they were connected to a local system. Telnet is defined in RFC 854 and will be elaborated in details in Chapter 3 as an authentication method used within Dynamic ACLs configuration.

Dynamic ACLs usually use Telnet as a method for remote user authentication. However Telnet is characterized by number of security disadvantages that don't guarantee a fully secure user authentication, and thus, the use of Telnet for the user authentication process of Dynamic ACLs is always prone to expose organizations' inner resources to many security risks and breaches. Due to these reasons, our study has taken place to evaluate Telnet as authentication method, to elaborate its pros and cons, and to suggest other

[2] Inner resources refer to the resources located on an organization network. Such a network is supposed to be securely guarded against security risks, including unauthorized access.

alternative authentication methods that can be used within the Dynamic ACLs configuration, in order to authenticate remote users. Thus, in Chapter 5 Telnet will be elaborated as an authentication method and compared to other authentication methods used within Dynamic ACLs configuration, such as authentication servers, including TACACS+, RADIUS, DIAMETER and Kerberos.

The study includes an important number of configuration codes that were specifically designed to support the security concepts analyzed and to provide guidelines for network decision makers to make better security choices that guarantee secure user remote access

KEY WORDS:

AAA server comparison, Access Control Lists, Authentication method lists, Authentication server, DIAMETER, Dynamic ACLs Authentication, KERBEROS, Network architecture, Network Security, OSI model, Proxy servers, RADIUS, SSH, Standard ACLs, TACACS+, TCP/IP model, Telnet, VPN.

INTRODUCTION

The research focuses on the user authentication step, which is the first step necessary to trigger the Dynamic ACLs filtering process and which involves the use of Telnet. So the research is to highlight the deficiencies of Telnet as an authentication method used within connections established by remote user devices, and filtered by one or more Dynamic Access control list(s), the so called Lock-and-Key security. As Telnet authentication is not considered a secure authentication by itself, the research interprets the details of such an authentication process while revealing all the security risks involved during such a connection.

The study focuses on detailing the remote user IP connections in particular, analyzing the details behind the functionality of Dynamic Access Control lists, as well as developing related configuration codes that show the use of the different authentication methods with Dynamic Access Control lists, along with elaborate explanations. The study will also focus on those authentication methods used within dynamic ACLs as alternatives to Telnet, as well as how they compare to Telnet as an authentication method, according to some comparison criteria, as a helping step on the road to overcome some of the Telnet authentication drawbacks and to facilitate the choice of different authentication methods used within dynamic ACLs according to different application contexts.

Throughout the study, the research approach proceeds by conducting an in-depth analysis of the related literature as well as the production of a number of thoroughly developed configuration codes that are specifically designed in order to support the concepts analyzed, on one hand, and to investigate the details behind their operation in different security context, on the other hand. Such analysis, code developing and tracing aim at the deduction of a number of suggested recommendations and best practice tips that the study will provide as valuable outcomes resulting from such a thorough research work.

Since the research deals exclusively with the security context, which is a very sensitive, secretive issue for any organization, it was impossible to gather practical data or configuration codes from any organization in order to support our study; this is why our research doesn't include practical data gathering, as could be expected from this type of research.

Also since the study's context is in the networking field, which is known for its dynamic nature that doesn't allow a downtime for testing, it was impossible to practically implement the codes that we have developed throughout the study on an actual working network, in order to verify their proper operation. Given the very high cost of building a whole new network from scratch only to help verify our configuration codes, that option was not available for us. These are the reasons why there were no means for the research to include practical experimentation through the results of which, a number of recommendations could have been developed. Also the same reasons lead to the fact that the configuration codes provided by the study, though thoroughly developed to support the concepts introduced and carefully designed according to the provided literature, were not implemented before on a real working network. Thus we recommend to anyone embracing our configuration codes to verify the codes in a real networking environment, before implementing them into an operating network in order to avoid unnecessary network complications.

The main objective of this study is to evaluate, analyze and critique Telnet as an authentication method, Dynamic ACLs as an access filtering concept as well as comparing their use benefits (combined together) to the use of other security approaches like proxy servers and authentication servers, namely RADIUS, TACACS+ and Kerberos. The study aims at providing a list of recommendations that will help guide for the decision maker to make better choices about the most appropriate security approaches that will guarantee his network secure user remote access.

Specifically, the goal of the study is to reflect on the remote access security subject, focusing particularly on the importance of the authentication process within Dynamic Access Control lists (ACLs) as an authentication means for remote users using IP-connections, and questioning the security reliability of Telnet as an authentication method combined with Dynamic ACLs. Also the research will investigate the use of alternate methods for authenticating the remote hosts attempting to access an organization's network through Dynamic ACLs, methods such as Secure Shell (SSH) and AAA servers including TACACS+, Kerberos, RADIUS, and DIAMETER, during the Dynamic ACLs filtering process. Besides, the research study will explain the drawbacks of Dynamic ACLs as a

security technology, introducing some alternative technologies and solutions in order to overcome these drawbacks.

The research contribution emphasizes the importance of the authentication process within Dynamic ACLs filtering, while focusing on Telnet as an authentication method usually used within that context, demonstrating its drawbacks and the security risks they represent as well as searching alternate authentication methods for Dynamic ACLs configuration that would overcome such drawbacks. The study includes different developed configuration codes, explanations and comparisons between the different authentication methods while emphasizing their benefits and, application as well as security limitation. Thus the study should be a helpful guideline for different organizations, as well as research groups, allowing them to easily compare as well as choose a suitable authentication method or combination of methods, according to their security policies and business needs, while configuring their border routers with dynamic ACLs for user IP access.

Thus, the research will cover the following subjects:

- In Chapter 1, we start this research by an in-depth explanation of the fundamental layers of network architecture, which are essential for any set of networked wires and equipment to establish a successful connectivity. We introduce both models: the OSI network architecture model and the TCP/IP network architecture model, describing all corresponding layers and specifying our layer of interest, the one where the authentication process takes place.

  Then we introduce the meaning behind the word "security" as a concept that protects every organization's network connected to the Internet, interpreting the different types of Internet threats and the different types of security solutions that puts end to such threats.

- In Chapter 2, the study explains in details the different types of ACLs, how they work, and their different application contexts, along with a number of configuration codes and examples that are specifically developed during the research in order to further support such an explanation.

- In Chapter 3, the study presents Dynamic ACLs mechanisms in details, along with some carefully designed configuration codes and examples demonstrating the authentication methods that are frequently used in combination with Dynamic ACLs implementations, and how they work.

- In Chapter 4, the study will present the most popular authentication methods used with remote user connections as well as their application within a peer recovery approach, along with a number of illustrating figures and configuration codes that are specifically developed during the research in order to facilitate the explanation and analysis of the different concepts introduced.

- In Chapter 5, the study will examine and analyze the differences between the authentication methods considered, their cons and pros compared to Telnet, on one hand, and in regards to their efficiency when used within dynamic ACLs, on the other hand. Again, accompanied with the different concepts analyzed and critiqued, there exist a number of developed codes, specifically designed to support the analysis within the chapter.

CHAPTER I

NETWORK SECURITY

Dynamic ACLs, the main subject of the study, are considered as a security solution to protect the resources on the organization's network/premises, thus the subject of network security is one of the core subjects in our study, as this introducing this subject will help the reader understand the importance to Dynamic ACLs as a security solution.

However, creating a secure connection to the organization's resources requires the connectivity establishment of such a connection, since the security can be considered as a further step to take place once all the network devices are correctly connected and up and running.

Therefore, before we start our study introducing details about Network Security, Dynamic ACLs, and User Authentication methods, let's first talk about the different layers of a network architecture that would allow any sort of communication to take place.

The network architecture will help us understand the detailed steps taking place during the user authentication process and will help us better understand them and relate them to each other.

The architecture will also help us compare different user authentication processes and relate them to their corresponding network layers, so that we can better understand their advantages and their drawbacks; and so we can substitute some of them in correspondence with their order and their position in the architecture.

1.1 The network architecture reference model

Over the years, many companies/organizations have created different their networking protocols and standards. However an open, vendor independent standardization, is a better model to show the networking architecture. There exist two open reference standards, which are:

- The Open System Interconnection (OSI) model;
- The TCP/IP model

1.1.1 The open system interconnection (OSI) model developed by the international organization for standardization (ISO)

The OSI model classifies all the protocols needed to establish a network connection into seven layers, according to the respective order of the processes taking place during a connection.

Each layer's input is basically the output of the layer below it; and each layer process initialization depends on the processes success and the completion at the lower layer.

Therefore if at one point, the processes supposed to take place at a certain layer fail or become unable to get to completion, there is no way for the upper layers to get any connectivity.

The OSI model layers are introduced as follows (starting at the very first layer where the network connectivity should take place):

1- Layer 1: The Physical Layer:

This layer refers to the standards of the physical characteristics of the transmission mediums, the rules concerning bit transmission and the rules concerning the transmission activation. This layer also refers to the definition of the physical electrical, optical connectors, pins, cables, voltage levels, electrical current, encoding or light modulation.

2- Layer 2: The Data Link Layer:

This layer refers to the rules permitting to a certain device to send data over a medium at a specific time and the rules defining the right format of the data transmitted (into frames). This layer provides means to recognize transmission errors as well.

3- Layer 3: The Network Layer:

This layer refers to the logical addressing of network devices to help communication over the network; the routing of network packets through the use of routing protocols that help find all possible routes to send a packet to its

destination; and the path determination process that helps network devices related to this layer (typically routers) to find the best way (route) sending a packet to its ultimate destination.

4- Layer 4: The Transport Layer:

This layer refers to the services related to the delivery of data to the destination, including flow control, error recovery, connection establishment and termination and data segmentation into smaller portions to facilitate transmission.

5- Layer 5: The Session Layer:

This layer refers to the management of the bi-directional flows between the two endpoints of the communication. Thus, this layer refers to the rules concerning how the connection sessions start, end, and are controlled; so that the upper layer, which is the presentation layer, has a transparent view of one continuous stream of data, even if it has been transmitted over many sessions.

6- Layer 6: The Presentation Layer:

This layer refers to the negotiation of the data formats (ASCII, ABCDIC, Binary, etc.); and data encryption.

7- Layer 7: The Application Layer:

This layer is mainly considered as an interface between the local applications and the communication software that would allow these local applications to communicate with other applications outside the local computer. Also at this layer, are defined the processes for user authentication.

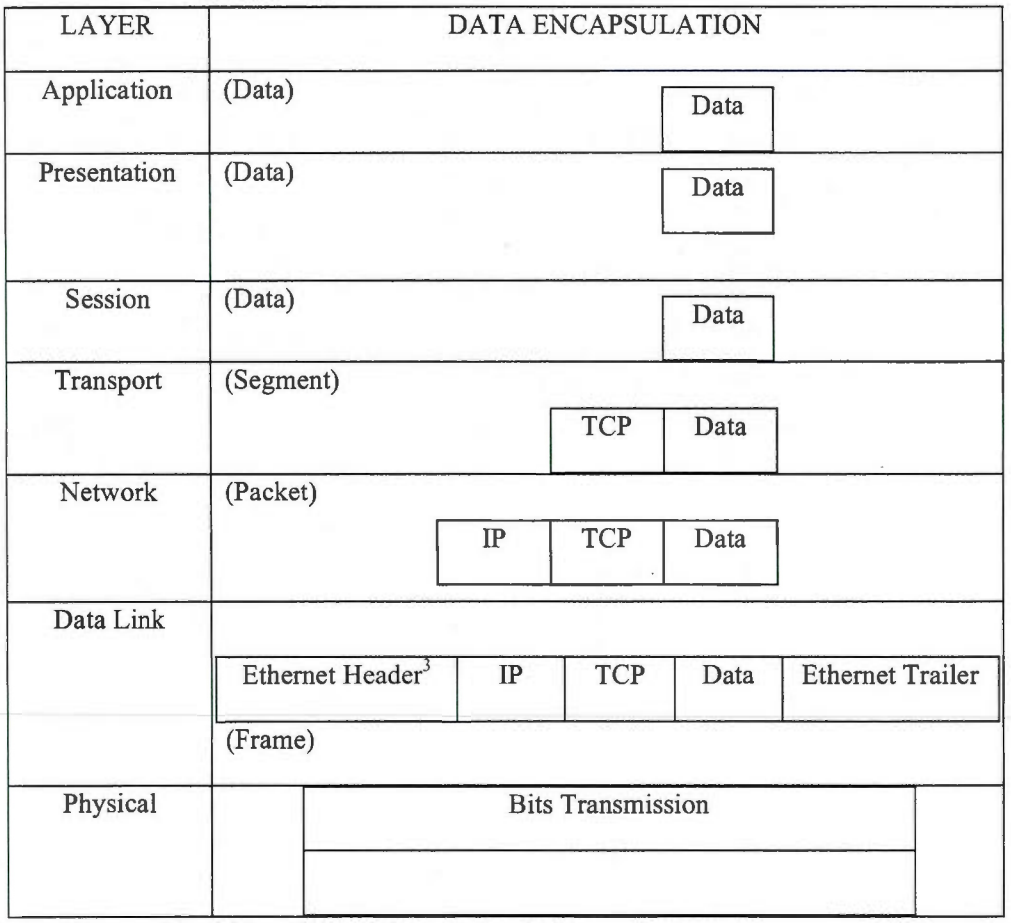

Figure 1.1 The OSI model and data encapsulation

As explained by Odom (Odom, 2009), in figure 1-1, every layer's data unit has a different name specific to the layer it belongs to. Also, every data unit coming from an upper layer, gets encapsulated into a header belonging to the next lower layer until it reaches the lowest layer (the Physical layer) where the data get encapsulated into a header and trailer for transmission.

Each layer provides a certain service to the next/upper layer. For example the Data Link layer helps the Network layer (the upper layer) transmit the Network layer packet using a layer-2 protocol. This process takes place without the Network layer protocol (layer 3

[3] Ethernet, as a header and a trailer, is demonstrated in table 1 as an example of a layer 2 protocol; other alternatives can be used like HDLC, PPP, Frame Relay, etc.

protocol, the IP protocol in this case) having to know the details about this transmission process.

Also, from the Data link layer point of view, what comes between the Ethernet header and trailer, which is basically the network layer packet, is only considered as simple data. Thus the Data link layer doesn't need to care about the details related to this data portion that corresponds to the network layer packet; as the Data Link layer has only to care about the transmission process of this Data portion.

The way the different layers of the network architecture work together emphasizes the importance of the encapsulation process that hides unnecessary details for each layer in the data payload, so that the layer can focus on its main job, only dealing with a given set of data. This breaks down the job related to the bigger goal of data transmission into small tasks that will be better performed when clearly defined, and assigned to each layer.

This break down helps to accomplish the tasks in a timely manner, while precisely helping to identify the reasons why a certain transmission might not take place, relating that issue to a specific layer malfunctions or alterations. It also helps monitor the errors occurring at each layer, and sometimes help for quality assurance.

The network architecture model helps classify layer protocols according to network link establishment needs and rules. And since each layer has its own set of protocols that corresponds to the tasks to be done within, the architecture provides a wide variety of alternative protocols choices, from which the best protocol can be chosen, for each layer, as it suits specific network design goals, scalability preference and performance perspectives. Thus the network architecture model, along with its layer protocols helps the network design process to meet its intended design goals.

Now that we introduced the network hierarchy of the OSI model, in the next section, we introduce the TCP/IP model, which is quite similar in concepts with the OSI model.

1.1.2 The TCP/IP model

This network architecture model has been developed by networking volunteers (Odom, 2009).

For the purpose of this study, we present the TCP/IP model and its different layers as a means of reference to help us understand the different steps of the user authentication process.

The TCP/IP model comprises a large number of networking protocols, which are documented in documents called Request for Comments (RFC). Devices that follow the implementation of the TCP/IP RFCs' protocols can easily connect to each other.

The TCP/IP model classifies networking protocols into four layers. We will illustrate them starting from lower to upper layers.

- Network Access layer: defines protocols defining data delivery over the physical media (electrical, optical) and rules for frames formats, using mediums for transmissions and transmission errors. It also requires the definition of the physical connectors, cables, voltage levels, and protocols for delivering data over WANs. Examples of protocols are Ethernet 802.3, HDLC, PPP, Frame Relay, RJ-45, EIA/TIA-232, V.35 and MAC protocols. Devices used on this layer are LAN hubs, repeaters, Switches, Wireless Access Points, cable modem, DSL modem.
- Internet layer: defines protocols providing logical addresses, routing and path determination. Examples of protocols are IP, IPX and SPX. Devices used on this layer are routers and multilayer switches.
- Transport Layer: defines protocols providing connection establishment and termination, flow control, error recovery and data segmentation for transmission. Examples of protocols are TCP, UDP, IPX and SPX.
- Application layer: defines protocols providing interfacing between the network and the different software application, authentication, data formatting organization and encryption and transaction flow management. Examples of protocols are Telnet, HTTP, FTP, SMTP, POP3, VoIP and SNMP. Devices used on this layer are firewalls and intrusion detection systems.

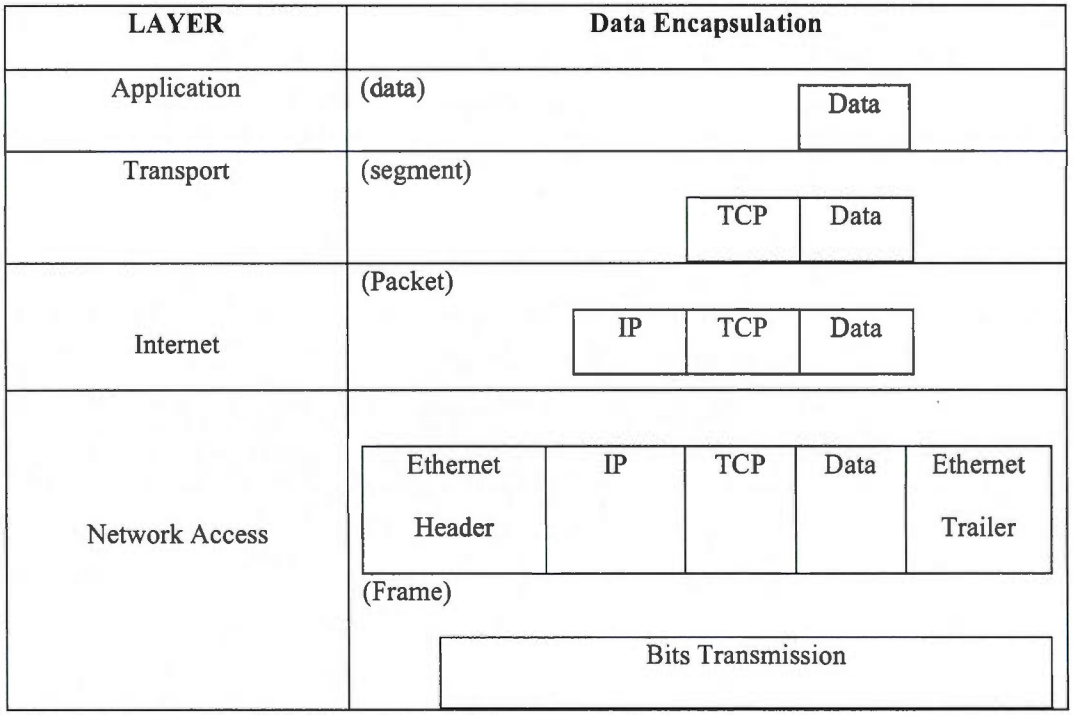

Figure 1.2 The TCP/IP model and data encapsulation

As shown in figure 1-2, and as in the OSI model, every layer's data unit has a different name specific to the layer it belongs to. Also, every data unit coming from an upper layer gets encapsulated into a header belonging to the next layer until it reaches the lowest layer (the Network Access layer) where the data get encapsulated into a header and trailer for transmission.

Again, the encapsulation process is used in the TCP/IP model for hiding unnecessary details from each layer as it provides a specific service to the next/upper layer.

Now that we introduced the network hierarchy through the TCP/IP model, we can proceed to the next main points of this chapter, which are the network security and the authentication process. While explaining the authentication process, we will refer to the TCP/IP model in many ways, so that we can have a better understanding of this process and how it works.

In the next section, we introduce the importance of network security and the types of threats an organization might face during its regular operation time. We will also introduce some security solutions that can help mitigate such threats.

1.2 The importance of network security

A couple of decades ago, few organizations around the world were permanently connected to the Internet, if even connected at all, as the Internet wasn't as popular and profitable as it is nowadays. These organizations were safe against illegal access attacks, as they were only using their inner networks with no exposure to outside security threats.

Nowadays, as the internet has turned to be an essential asset for many businesses and organizations, and a very important social and business communication means, all organizations become interconnected through the internet, and consequently, their inner resources become exposed to greater security threats.

In order to better understand security, let's get a general idea about the internet and how people and organizations can connect online, as it follows in the following section.

1.2.1. The Internet and the World Wide Web

According to W3 (W3 - WWW, 2009), the Internet is a global system of interconnected computer networks that interchange data by packet switching using the standardized Internet Protocol Suite (TCP/IP). Thus, the Internet is defined by the TPC/IP standards. The Web, on the other hand, as defined by W3 (W3 - WWW, 2009), is as follows: "The World Wide Web (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI)."

As exactly cited by W3 (w3.org - WWW), "The World Wide Web (known as "WWW', "Web" or "W3") is the universe of network-accessible information, the

embodiment of human knowledge. The Web has a body of software, and a set of protocols and conventions. Through the use hypertext and multimedia techniques, the web is easy for anyone to roam, browse, and contribute to. The World Wide Web began as a networked information project at CERN, where Tim Berners-Lee, now Director of the World Wide Web Consortium [W3C], developed a vision of the project. "

Thus, the Web is defined by other specifications. The first three specifications for Web technologies defined by Uniform Resource Locators (URLs), Hyper-Text Transfer Protocol (HTTP), and Hyper-Text Markup Language (HTML) (W3C).

A more specific explanation of the difference between the Internet and the Web is presented by the definition on Webopedia website (Webopedia - Web versus Internet, 2011); stating that the Internet is the network of networks, upon which any connected computer/device can communicate with its peer(s) as long as they are both connected to the Internet. Information are transmitted over the internet via languages known as protocols.

Thus the Internet is an infrastructure that allows access to the Web service and to other services, like the mail services, of a distant organization. (Internet Society, 2012).

On the WWW, organizations providing information or services have web servers that are interconnected among themselves, as well as among an endless number of remote hosts that access these servers to meet the users' needs. These servers deliver different services and information, for example, a web server provides web services and stores the web pages of an organization, a file server stores files for an organization, and an email server manages the emails within an organization.

In order to better explain the interaction between two Internet-connected devices, let's consider the case of a user/host connecting to an organization's Web server to get some information. The organization's web servers store the web pages that include the information needed by the web hosts. When a web host needs to access the organization web server, he acts as a web client and thus a number of application layer processes must take place:

- The web client has to use some software to enable him to connect to the organization's web server and thus, display the web pages; that software is the host's web browser.
- The connection takes place when the web client is able to specify the web server, the web page and the network protocol to get the information from the page.
- In order to specify the server, the host needs to get its IP address, either through the use of Domain Name System (DNS) or through static configuration of that IP address.
- Each web page stored on the organization server consists of several files that will be sent by the web server to the host and through the use of Hypertext HTTP.

In order for the hosts to meet their needs for services, they still have to access the organization's web servers through this same set of steps. Performing these steps demands the existence of network connectivity. However, transferring sensitive data, including financial information, has to go through electronic commerce applications (e-commerce), where data needs to be secured through the use of a certain security technique, such as Transport-Layer Security (TLS) application-layer feature, instead of regular unsecure HTTP.

1.2.2. The benefits of network security

The network security techniques and features provide some essential benefits for any network connection. These benefits are as follows:

- Availability: of the network resources needed either in order to establish the connection, or in order to obtain the service and information needed through the connection.

- Authenticity: of the remote user/device accessing the network, so that there would be a certainty that the remote user is exactly who he claims to be.

- Integrity: of the information contents being transferred over the connection, so that there would be a certainty that there was no replay/modification for these information.

- Confidentiality: of the information being transferred over the connection so that there would be a certainty that they were not revealed to non intended parties.

(Odom, 2009)

Now that we introduced the network security benefits, lets' highlight some of the network security common threats.

1.2.3. The threats to organizations' networks

Years ago, security attacks were commonly performed by nerdy students who needed to prove that they could break into an organization's network. Nowadays, malicious attacks don't need savvy knowledge, as spying tools are abundantly available and free for any person to perform any range of attacks he dreams about. This doesn't mean that newly reported attacks don't show a high-end degree of sophistication and subtlety.

This leads us to the conclusion that every single day, with every single new person and organization connecting to the Web, adds up to the vulnerabilities of all Web connections. This is true, especially if we consider that today the motive behind the attacks is much beyond a personal challenge, as it might be a criminal attack, a financial steal, a national espionage or even a destructive or terrorist attack against public services or government networks.

Security attacks can be performed from the inside of an organization, as well, through the unauthorized access of one of the organization's staff member to the inner network. This person could have the same exact motivations as an outsider attacker, as mentioned above. This would even increase the vulnerabilities of any organization's network, as well as its needs to apply rigid network security strategies to limit such attacks.

According to Odom (Odom, 2009), security attack types can be classified into the following categories:

1. Attacks compromising the availability of the network resources: like the Denial of Service (DoS) attack; its main purpose is to disable the hosts/data/software/network connections/network communication.
   DOS includes:
   - Crashers: attacks causing hosts failure or network connection failure
   - Destroyers: attacks causing damage to hosts, data and software.
   - Flooders: attacks flooding connections with a great amount of packets creating an unsustainable level of traffic for the network to make any connection useless.

2. Attacks compromising the confidentiality of the transferred data: like the access attack; its purpose is to steal confidential data.

3. Attacks compromising the authentication of the remote host: it takes place when hosts fake their identity for malicious purposes, such as collecting information about network resources in order to start another attack type.

Attacks can be performed using a wide range of tools that allow an attacker to defeat access policies. These tools can be classified as follows:

- Viruses, which are malicious programs that infect other programs, in order to cause problems or steal information. Protecting computers usually involves using an anti-virus program that recognizes the characteristics of known viruses so that the computer can avoid them amongst received packets, as well as within the file system during an anti-virus periodic scan. These anti-viruses have to be updated continuously with newer releases to enable a computer to cope with an ever-evolving virus population.

- Scanners, spy on the computers to get information about the network services and the operating system it uses, by sending connection requests to different applications, with different UDP and TCP port numbers. The spying action of a scanner is to gather information, which would usually help the attacker to reach his end purpose of damaging the software or the hardware.

- Spyware, which tracks the user activities performed on the computer and sends this information along with sensitive and private data to an attacker via the Internet.

- Worm, which is a program duplicating itself independently on the internet and propagating on organizations' networks, usually with the purpose of paralyzing the network traffic, and usually leading it to a DoS attack.

- Keystroke logger, which is a program that captures user credentials, by tracking the user keystrokes, especially when the user accesses a secure web site or enters sensitive data. Such keystrokes are recorded and reported by this program to the attacker, who will be able to exploit the stolen credentials.

- Phishing, which is a special case of Scamming, is about faking up a web site and making it appear exactly like a legitimate one, usually a financial company or bank. The attacker then sends this faulty web site, usually in an email, to the user and asks him to enter some sensitive information like his social security number and passwords. Unaware, the user might be tricked, giving the attacker the chance to steal his real accounts.
  (Odom, 2009)

In order to face such threats, an organization connected to the Internet, should apply many security protection mechanisms, including hardware and software, in order to allow the organization to defend its inner resources. The best-known mechanism is the

establishment of firewalls on the organization's perimeter/boundary to the outside world of Internet.

However, securing the organization perimeter isn't considered a full-proof guarantee since security attacks might come from the inside of the organization, as well as from the outside. Some examples of such a variety of attacks could be as follows:

- Access from the Wireless Local Area Network (WLAN): As the wireless signals might leave the physical building of the organization, an outsider attacker might capture these packets and get access to the organization inner network to perform malicious attacks.

- Infected laptop: As the user accesses the Web through an unsecure connection, typically from his home, the laptop might get infected by a virus, or another malware. Later, that user might return to the organization's building and, unaware of his laptop infection, he might bring that laptop to connect to the organization's network. This virus infection might get to the organization's network, and thus other PCs and devices all over the network, especially if not scanned with the daily anti-virus software scan, might get infected as well.

- Disgruntled employee: A user planning to move for another position in another organization might want to steal the whole database of the company, or perhaps sensitive data on a Flash card or an MP3 player, very discretely to carry outside the organization's building. This stolen information might be used later for even bigger crimes.

- Rogue Dynamic Host Configuration (DHCP) Protocol servers and rogue routers for IPv4 and IPv6: As inner routers and switches learn their IP addresses, and other connectivity information like Domain Name System (DNS) Servers from their Default gateways, which are their DHCP

servers/routers, attackers might setup rogue servers/routers to appear as if they were legitimate ones, then the other network devices will communicate information to these rogue servers, allowing them to control the network, and to allow the attacker to manipulate his malicious plans. "In effect, this becomes a type of "man-in-the-middle"; the attacker is wedged into the path and the client doesn't realize it." (Hucaby, 2010)

- Address Resolution Protocol (ARP) spoofing: ARP works when a host needs to know the layer 2 address (MAC address) of another host whose IP address is known. As the attacker might craft a MAC address that appears as a legitimate one, the legitimate hosts on the network might trust that MAC address and start sending traffic to the rogue host.
  So the attacker's host will be right in the packets path of the legitimate network, which allows it to intercept legitimate packets and their contents. "This attack is known as ARP poisoning or ARP spoofing, and is considered to be a type of man-in-the-middle attack." (Hucaby, 2010)

- Spoofed IP addresses: Hosts on an inner legitimate network are supposed to use the IP addresses assigned to them, in all sorts of traffic. Attackers, however, can spoof some IP address and use them within the organization's inner network. This spoofing might take place through borrowing some IP address of legitimate network hosts, or by using IP addresses at random. When the rogue host sends traffic to the legitimate host, the legitimate host will not find the destination to send back the traffic to since that destination would be represented by that rogue IP address that was the original source of the traffic. Thus no traffic will be returned to the rogue originator, who can then easily start network attacks (like DoS attack).

- Unauthorized users connecting through an authorized connection: As an authorized user connects to the network, he might use a weak authentication method like telnet through its authorized connection. Such authentication

might result into another unauthorized user being able to sneak into the connection, spying and stealing the authorize user's credentials. These credentials will enable this attacker to connect through the same authorized connection into the organization network and perform further harm.
Actually, this kind of attack is the focus of the study, as we will see later in the upcoming chapters.

The previous attacks are only a few examples of ways people can connect maliciously into an organization network. Fortunately, there are many ways organizations can take action in order to avoid these attacks as much as possible. This will lead us to the upcoming section elaborating security in general.

1.2.4. Security actions taken by the organizations to protect their networks

Security is considered one of the most important issues when building the network topology of any organization, especially since security mechanisms, including software and hardware have to be continuously evolving and updated in order to increase their ability/performance to cope with the ever evolving malicious techniques appearing all over the internet.

Let's first define the term "network security", starting with a couple of definitions from Wikipedia (Wikipedia - Network Security, 2012):

- "In the field of networking, the area of network security consists of the provisions and policies adopted by the network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of the computer network and network-accessible resources."
- "Network security involves the authorization of access to data in a network, which is controlled by the network administrator."

Most organizations use a comprehensive security plan in order to react to the different types of threats mentioned above. "Anti-x" is the name of a wide range of security tools that can save the enterprise inner network the damage, time, money and effort facing such attacks. These tools include the following:

- Anti-virus: scans network traffic preventing the transmission of known virus based on virus signatures.
- Anti-spyware: scans network traffic preventing the transmission of spyware programs.
- Anti-spam: scans emails before they reach the user, deleting or segregating junk emails.
- Anti-phishing: looks for rogue URLs sent in messages, preventing phishing attacks from reaching the user.
- URL filtering: filters Web sites URLs preventing web users from connecting to inappropriate sites.
- E-mail filtering: acts as an anti-spam tool and filters emails that include offensive materials.

(Odom, 2009)

Besides these security tools, the most common security protection mechanism an organization uses, as mentioned above, is firewalls installed at the perimeter of the organizations network, hence on the boundary between the organization's inner network and the internet. Adaptive Security Appliances (ASA) can act as firewalls, or in combination with other security roles, to help protect the inner network of an organization.

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are other types of security tools that help organizations keep their networks safe and intrusion free. IDS are security functions that are situated on monitoring ports, and that passively compare network traffic patterns to a list of well-known attack signatures and characteristics. IDS may also rate each threat and report it to other security devices, like firewalls and routers, which would help such devices mitigate such attacks.

Intrusion Prevention System (IPS), which are security functions sitting in the packet's path, act almost exactly as IDS, rate the threats and react to them, however unlike IDS, IPS actively save the network by actively filtering the offensive traffic.

Virtual Private Network (VPN) tunneling refers to the process of securing the traffic between two devices while crossing the Internet. This process includes packets encryption and encapsulation, as well as endpoints (users) authentication. VPNs will be further explained in appendix A.

Cisco has introduced a whole range of integrated security mechanisms represented in one security model called "Security in Depth". This security model includes tools that can work in harmony with other security features found in routers, switches and other devices all around the network, by automatically reacting to network problems and attacks. This model is sometimes referred to as a "Self-defending network". (Odom, 2009)

One example of such automatic tools is the Network Admission Control (NAC), which enforces the organization's network infrastructure to make sure all devices attempting to gain access to the organization network are complying with its security policy. NAC detects the moment when a device/host starts a connection into the organization Local Area Network (LAN); it recognizes the user, his devices and his network roles; it evaluates if his device is compliant with the organization security policy; and prevents that connection to the LAN unless the host's antivirus definition is updated, its full antivirus scan is performed and the user authentication is checked (by entering a user name and password). In other words, the host gets blocked by the NAC, in case his connecting device is not compliant with the organization security policy. In this case, some alternatives are offered to that user such as getting connected through a guest access link, to the organization network. NAC audits and reports the connected devices on the network as well (Cisco - Cisco NAC Appliance).

Now that we explained the network security establishment as well as the types of security threats an organization faces along with some of their mitigation solutions, let's

introduce the authentication process[4] that a user goes through in order to access the organization's inner resources.

1.3 The location of the authentication process on the OSI model

The authentication process takes part of the security process in general, and its main role is to identify whether the user attempting to access the network is who he claims to be. Thus the authentication is very important for an organization as it gives it means to specify whether a given user can be allowed to access its inner resources, or denied that access.

Now, let's try to locate the user authentication process on the layers of the OSI model. Actually the user authentication process takes place at the seventh, and last, layer of the OSI model, namely the Application layer (Odom, 2009), which is the layer that interfaces with the user when he begins an attempt to start a connection session into the organization's network devices. Usually these devices are firewalls that separate the organization's inner network from the open unsecure Internet. Upon these firewalls, the dynamic ACLs as well as other ACLs and security solutions will be configured in order to take action filtering the illegitimate users traffic whenever it comes in.

Other authentication processes different from the user authentication, might take place at different layers, or might not take place at all, as it is the case with host-based authentication which is about authenticating the remote device (host) rather than the user (using 802.1X user authentication). This authentication encompasses an Open SSH feature that, when enabled on an untrusted device, allows that device to be accessible by the user without having to enter his credentials. Thus, though the authentication doesn't take place once the user accesses the device, it is set in advance through the layer 7-Open SSH (Computer Emergency Response Team, 2011).

Another example where the authentication takes place on a different layer is the MAC-based authentication[5], which is about authenticating the layer 2-source MAC address

[4] The authentication process will be explained in details in chapter 4, entitled "User authentication".

of the remote device rather than the user. That is an example of an authentication that takes place at layer 2, since that is the layer corresponding to the MAC addresses (Allied Telesyn - How to configure MAC-based port authentication, 2005).

We will also notice that the encryption, which is a process that usually takes place as part of the authentication process, takes place at the sixth layer of the OSI model, namely the presentation layer (Odom, 2009), which is responsible for setting the different coding of the transmitted data. This happens in case of using a layer 7 application to communicate with a remote device in the form of same layer connection. However, in other cases, the encryption process can take place at layer 2 or 3 depending on the corresponding layer of the used connection protocol. For example PAP and CHAP encryptions[6] take place at layer 2 since the corresponding connection protocol is the layer 2 Point-to-Point protocol.

Since the study's main focus is the authentication process that takes place during the use of Dynamic ACLs as a security solution, the application layer will be the layer of interest for our research. Dynamic ACLs depend on Telnet as a connection method, as we will explain in chapter 3. And since Telnet is an application that takes place at layer-7, the application layer (Odom, 2009), then all our research will be about the application layer.

Though we will introduce some encryption concepts as well as other Transport layer concepts as needed throughout the study, it is important to pinpoint to the reader that the application layer is the layer of interest for the research study.

In general, chapter 1 can be considered as the necessary base upon which further analysis is built during the whole research. The chapter studies in details the main concepts of network architecture, connectivity, security and authentication. This detailed study is based upon a thorough analysis of the theoretical information obtained through the number

---

[5] MAC address-based authentication is not considered as secure authentication method due to the ease of its mitigation using a MAC address sniffer, which facilitates MAC Address spoofing (Superuser). The MAC Address spoofing will be addressed in chapter 5, entitled "Authentication and Dynamic ACLs".

[6] PAP and CHAP encryptions will be explained in details in Chapter 4, entitled "User Authentication".

of references that was used. The analysis actually provides concise pieces of information that will serve the reader to understand further concepts that will be introduced, critiqued and analyzed in the following chapters.

In details, the chapter starts by an in-depth explanation of the network architecture models (OSI and TCP/IP model), then it follows by introducing the importance of network security as an aspect that affects the interconnected networks, its benefits for any network as well as the threats it can eliminate. Finally, the chapter ends with information that link the authentication process, which is the main focus of our study, to the network architecture model, in order to help the user identify the network layers of interest for our study.

All in all, the chapter helps lead the reader to understand more detailed concepts as the ones introduced in the following chapter, concepts about security ACLs that the organization can implement on its firewalls to ensure illegitimate access mitigation, their mechanism, different types and different implementation needs.

CHAPTER II

ACCESS CONTROL LISTS (ACLS)

2.1 What are ACLs?

In order to explain the concept of ACLs, let's first consider the example of an organization's network that needs to communicate with the outer world, in order to share information and to perform e-commerce. Such a network needs to be well secured against all sorts of outer threats described in chapter 1.

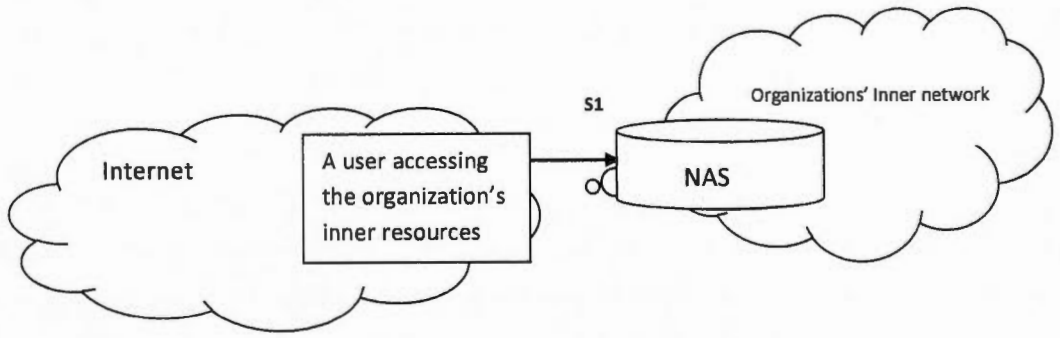

Figure 2.1 The NAS and the inner network security

In order to avoid such threats, the organization has to plan/design its network defense techniques adequately. Several defense techniques were introduced in chapter 1, however the most commonly applied security design includes a Network Access Server (NAS), which is a router situated at the border of the organization network, and is acting as the main gateway to communicate with outer networks, as shown by figure 2-1. NAS is

typically more than just a terminal server[7], as it may support protocol based access services, such as Point-to-Point (PPP), Apple talk Remote Access Protocol (ARAP), and others.

The Network Access Server (NAS) is a border router/firewall, whose Command Line Interface (CLI) can be accessed through user Exec access mode, or Privileged access mode. The Exec access mode allows remote users to use the router as a connection means to the organization's network. The Privileged access mode allows remote users to use the router for administration purposes. The Privileged access mode can only be reached after successfully logging into the Exec access mode. The router modes as well as the detailed processes related to their access will be described in details in chapter 4.

The NAS is usually configured using tools like Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), as well as other tools like Access Control Lists (ACLs), to help protect the inner network. Such tools will prevent inbound unwanted/unauthorized IP packets from entering the inner network usually by discarding them, while permitting network access to authorized IP packets. Inbound packets are specified to go towards the inner network, therefore, they're said to have the direction (in).

These tools also protect the network by allowing outbound packets to reach some specified destinations, while preventing them from reaching other unsafe destinations. Those packets are specified to go towards the outside world, therefore, they're said to be outbound, going (out) of the inner network.

ACLs are considered one of the security means that, when configured on a network device, can help that device to filter the IP packets it sends or receives according to security decisions made by the network engineer. Such security criteria and decisions will be included in the ACL as defined by the engineer.[8] Since the most commonly applied security

---

[7] Terminal servers usually provide the remote users with a character mode front end and allow the user to Telnet or rlogin to another host on the network.

[8] Throughout this study, we will refer to these conditions using the words "conditions", "statements" or "rules".

design refers to the application of ACLs on Network Access Server (NAS), we will explain the ACLs accordingly for the rest of our study.

Thus ACLs can be explained as a group of rules and conditions, dictated by the network engineer, configured on a border router/firewall (NAS) that puts these rules into action once the corresponding conditions are fulfilled.

"The goal of these filters is to prevent unwanted traffic in the network-whether preventing hackers from penetrating the network or just preventing employees from using systems they shouldn't. IP access lists can also be used to filter routing updates, to match packets for prioritization, to match packets for VPN tunneling as well as to match packets for implementing Quality of Service (QoS) features. ACLs can also be used as part of configuring Network Address Translation (NAT), (Odom, 2009)."
"ACLs might also be used for filtering non-IP protocols such as AppleTalk or IPX." ((Cisco - Auth Proxy, 2007)

In other words, Access Control Lists (ACLs) are bodies of network configuration commands applied usually on an organization boundary device (such as a firewall, Cisco Adaptive Security Appliance (ASA), Network Access Server (NAS) or a Default Gateway).

The benefits of ACLs can be summarized in the following points:

- Using ACLs can eliminate some traffic, which can increase network performance.
- Using ACLs helps restrict the delivery of routing updates, which allows flow control for network traffic.
- Using ACLs controls the type of traffic sent or received by the hosting device upon which they're configured.
- Using ACLs controls the areas that a client can access, within a typical client-server environment.

  (Orbit Computer Solutions, 2012)

The ACL mechanism proceeds in two steps:

1. Matching: it means to watch the IP traffic and to look for certain criteria about that IP traffic; once that traffic matches these criteria, specified in the statements of the ACL's body, a match is made and execution proceeds to step two.
2. Action: it means the action specified in the ACL takes place. This action is either to stop the traffic, thus the IP packets will be discarded; or to allow that traffic, and thus these IP packets will be able to continue their way towards their destination.

Actually the router on which the ACL is configured will act like a guard for the inner network of the organization, as it will decide about permitting each packet to go through or to get dropped, by matching the information, included inside the packets headers, with the conditions in the body of the ACL. This filtering is thus made at the network layer of the TCP/IP model described in the pre-introduction chapter of this study.

While configuring an ACL, the engineer has to make some important decisions: after specifying the network resources or portions that should be protected, as well as the unsecure packet's threats that should be stopped (filtered) by the ACL, the engineer has to determine the characteristics of the packets that should be filtered, whether they are incoming packets or outgoing ones. He also needs to determine where the ACL should be enforced, that is, at which specific interface on a specific NAS/router/boundary device the ACLs should be applied.

Actually, IP packets can be filtered by the ACLs as they enter the NAS interface, thus before routing can be made by the router. This is called inbound packet filtering. IP packets can also be filtered by the ACLs as they exit the NAS interface, and that is called outbound packet filtering. Since both processes: the routing process and the IP packet filtering take place at the network layer (layer 3) of the TCP/IP model, there is no difference in the order they occur, for the organization's inner network, architecture wise (whether the routing take place first of the filtering takes place first). However, performance and security wise, the order of these two processes, might be of a certain importance and significance for

the inner network they are meant to serve. This order is different according to the type of ACLs implemented on the NAS router, as we will see in the following sections.

2.2 IP ACLs filtering process

The syntax of ACL's filtering rules is as follows:

- If the IP packets are to be stopped/ filtered by the ACLs, then the Word "Deny" must be used in the configuration, which implies denying access by these filters. Likewise, if the IP packets are to be allowed access by the ACLs, then the Word "Permit" must be used in the configuration.

- The filtering logic, which includes all the conditions that should be matched with each IP packet in the passing traffic, is configured using statements making the body of the ACL.

- An implicit "deny all traffic" rule is enforced after all the specified rules, which means access is denied for any IP packet not conforming to the conditions listed in the body of ACL. For best security practices, it's highly recommended to explicitly configure one last statement, to the same effect, as it will give an accurate indication of the number of ACLs violations that have taken place during IP traffic, to the network Administrator/Engineer.

  The ACLs work as follows:

  1. The first condition, situated in the first statement in the body of the ACL, is compared against the packet's characteristics in the network traffic;
  2. When a match is found, the action configured in the ACL takes place, either permitting or denying traffic.
  3. If no match is found, step 1 and 2 are repeated for checking a match with the following conditions until a match is found.
  4. If after going through all the conditions, still no match is found, then the traffic will be denied (Odom, 2009).

2.3 How to configure an IP ACL?

Before we consider the ACL's configuration topic, let's explain an important detail about the ACL's filtering process: The filtering process specified by the ACL depends on wildcard masks. The wildcard mask defines the range of the IP addresses of network traffic, that will go through the examination and filtering process. All other traffic will be automatically denied without examination. Thus, within the ACL's configuration, both the network traffic, as well as the range of this traffic to be examined, will be identified by a network IP address as well as a wildcard mask that identify the packets of interest, respectively. So, in order to match all hosts in a certain subnet[9], all we have to do is to invert this subnet mask and all the traffic that will be eligible for filtering will represent all hosts within that subnet; any other host outside the subnet will be denied access without going through the filtering process. For example, in order to match all hosts of a subnet with a subnet mask 255.255.240.0, the wild card mask that should be configured in the ACL would be 0.0.15.255.

Thus, this simple process takes place by specifying the address octets to be filtered. A bit value of 1 means a "don't Care" bit, while a bit value of 0 means an "exact match" bit. Thus, in the previous example, the first two octets are all "exact match" bits and the last octet is all "don't care" bits. In order to match a specific host IP address, the wildcard mask of 0.0.0.0 can be used, referring to the fact that all four octets are "exact match" bits (Odom, 2009).

Using the same concept of wildcard masks, only certain hosts or certain subnets can be allowed access, while others that belong to the same network, might be denied access. A full explanation of the wild card mechanism is not the focal point of our study, so we will not go into more details about this subject.

---

[9] Subnet is a smaller part of a bigger network.

2.4 IP ACLs categories

In order to better understand the configuration of IP ACLs, let's first state the different types/categories of ACLs, as the configuration depends on each type functions and features.

2.4.1 Standard ACLs

- A Standard ACL uses a simple logic, as it matches only the source IP address of the traffic to be filtered. Standard ACLs can be configured to match the whole packet source IP address or only a part of it, as mentioned earlier.

- Each standard ACL is associated with a number. Standard IP ACLs numbers range from 1 to 99 and from 1300 to 1999. All ACLs statements associated with the same number are considered within the same ACL, and they will be processed in the order they are stated within the ACL's configuration.

- All statements (having the same number) will be listed within the ACL, and will be executed by the Network Operating System, in the same order as they were placed within the ACL configuration. The statements are matched sequentially, using first match logic. Thus once a match is made for a certain IP packet, the search is over and no more subsequent Standard ACL's commands are to be matched.

- All conditions or fields stated on one ACL statement must match a packet in order for that packet to be considered a match for that statement.

- When an ACL is globally[10] configured on a device, all the statements associated with this ACL (having the same number) start with the "access-list" commands. However, when the ACL is configured on a specific device interface, the associated statements will be first globally configured on the device starting with "access-list"

[10] Globally configured ACLs are configured on a device global configuration, versus a device specified interface. Usually the device is a Router or a layer 3 Switch.

commands, as well as configured on the interface starting with the "ip access-group" interface subcommand while referring to its global configuration once using the ACL number

- The position of the ACL on a specific router interface has to be carefully chosen, since the ACL's security performance differs according to its position regarding the source IP address to be filtered, or the Destination IP address to be protected. That's why the planning of the location of the Standard ACLs by choosing a specific inner router's interface to configure the Standard ACL is important.

  Briefly, standard ACLs should always be enforced near the packet's destination, which is the specific network resource to be reached. Configured otherwise, the Standard ACL might discard the allowed packets unintentionally.

- Once the best interface location is determined, the standard ACL can be enabled on the interface using the **"ip access-group"** command.

- Since a Standard ACL only examines Source IP addresses, the direction specified in the ACL has to be mentioned in the interface subcommand configuration. Specifying the direction of the IP packets examining process permits the matching of the IP packets as they go in the direction that the Standard ACL is examining. The direction is either "in" or "out". Globally configured ACLs statements don't have to include a direction keyword.

- The generic syntax of a Standard ACL configuration command is as follows (Odom, 2009):

  Access-list *standard-ACL-number* **{deny|permit}** *source-IP-address Source- Wildcard*

  If the source of the IP traffic to be examined and matched is only one IP address, rather than a whole network or subnet, then the wildcard can be omitted from the

Standard ACL's configuration, and the word "host" can be written before the "source-IP-address" portion, within the ACL's configuration.

Example 2-1

Example of standard ACL

In this example, we start by illustrating a network topology, followed by defining the security needs, and the configuration-code of the standard ACL that will help meet such needs. Then, we follow with the explanation of the configuration code, and finally, with the explanation of the process behind the code to help the reader better grasp the concept behind the way how standard ACLs work.

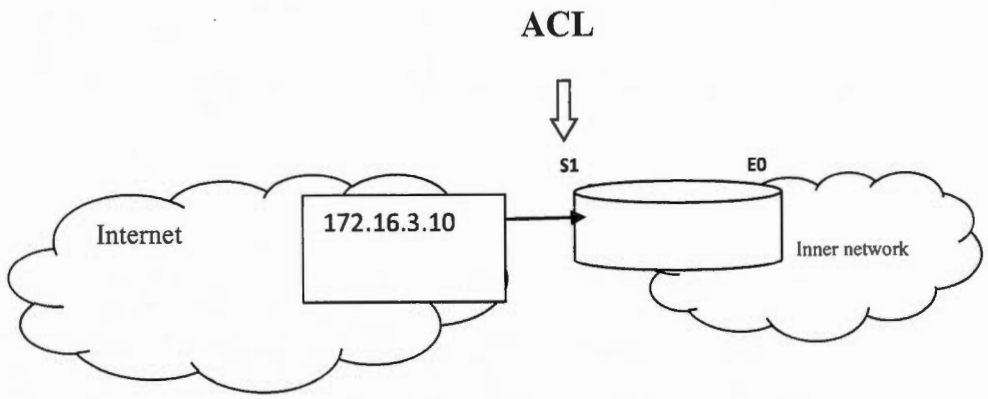

Figure 2.2 Standard ACL configured on interface E0

Consider the network topology illustrated in figure 2.2. The organization needs to prevent IP packets coming from the source IP address 172.16.3.10 from accessing its network, while allowing access to the IP packets coming from any other source.

The configuration will be as illustrated in figure 2.3:

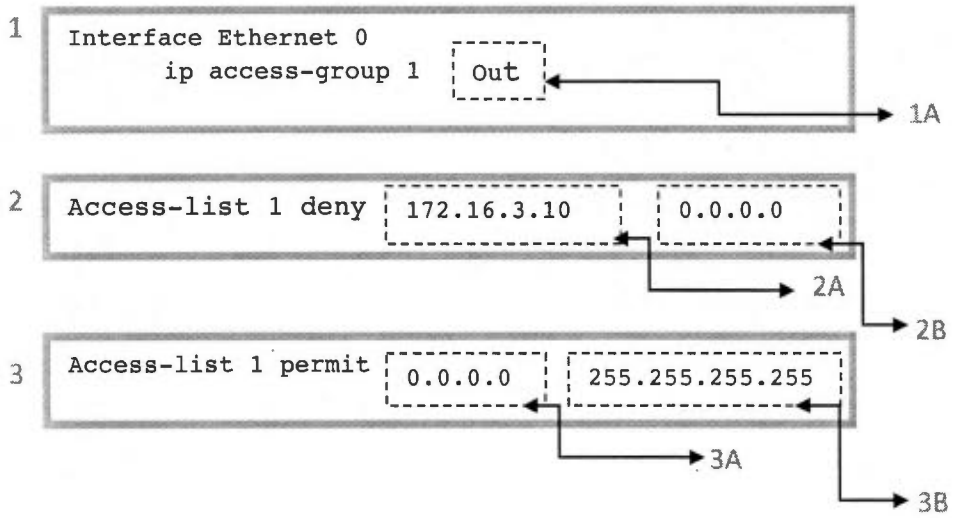

Figure 2.3 Standard ACL configuration code on interface E0

The configuration code of figure 2-3 can be explained as follows:

- The statement in part 1 of the figure is not part of the ACL itself. This statement specifies that the access list is applied to the router's interface. The ACL is applied to the Ethernet interface (E0) (rather than the serial interface S1), which is the interface closer to the inner resource to be protected. That resource is, then, considered a destination since the traffic to be examined flows towards the inner network.

  This traffic is flowing out of interface (E0), towards the inner network, hence the configuration word "out" in part 1A. So the Standard ACL will examine the traffic as it goes out of the router's E0 interface, on their way onto the Ethernet link to reach the organization's inner resources.

- Part 2 of the figure is where the standard ACL itself is defined, globally on this router, rather than applying it exclusively on a specific interface. Any packets

coming from the source IP address 172.16.3.10, mentioned in part 2A of the figure, will be stopped and discarded by the ACL. The wildcard mask 0.0.0.0, mentioned in part 2B of the figure, means all octets matter in this match, which means performing an exact match for that exact source IP address.

- The last command, in part 3 of the figure, is to allow access for any other source IP addresses. The wildcard mask 255.255.255.255, in part 3B of the figure, means all octets are to be considered as "don't care" bits for any match to be performed by this Standard ACL. The IP address 0.0.0.0, in part 3A of the figure, refers to any IP address on the internet, as long as it is different from the denied IP address mentioned in part 2 of the figure.

After we have analyzed the standard ACLs' configuration code, let's explain the process that takes place behind it.

The standard ACL's body here as we saw in part 1, starts with applying the ACL to a router's interface so that the filtering process takes effect, since configuring the ACL statements globally on the router doesn't start the filtering process. Next, the ACL statements global configuration includes two conditions (mentioned in part 2 and part 3 of the figure). When performing the matching for the outgoing IP packets, the first condition is tested first, if no match is found, the second condition is tested. If still no match is found, an implicit deny all is applied to discard any traffic that doesn't match any of the sequential conditions in the Standard ACL.

Now that we introduced the Standard ACLs mechanisms, it will be easier to understand the Extended ACLs mechanisms, upon which Dynamic ACLs are built. Thus a thorough understanding of these ACLs will help us to reach our goal of this study.

2.4.2 Extended ACLs

Why do we need Extended ACLs ?

Standard ACL can deny one host from accessing an inner server 1 while denying the others. However, this standard ACL cannot deny access to server 1's from that host, while allowing that same host access to another inner server 2.

In order to do that, an extended ACL is needed to allow the organization's network engineer to have control over both the source addresses that would like to access the organization's network, on one hand, and the destination IP addresses that correspond to the network inner resources, on the other hand.

How do extended ACLs work?

- Extended ACLs use more sophisticated logic than standard ACLs, as the filtering process includes, not only matching the source IP addresses, but also matching the destination IP addresses, along with source and destination port numbers. The protocol used for transmission is matched as well, so that extended ACLs can match IP or ICMP protocols (layer 3 protocols), as well as TCP or UDP protocols (layer 4 protocols) (Odom, 2009).

- The port number matching use basic operations like in table 2-1:

The operator	What it means
neq	Not equal to
lt	Less than
gt	Greater than
range	Range of port numbers

Table 2.1 Extended ACLs operators

➢Some port numbers are well known to represent certain applications they connect to, and they must be used when connecting to these applications. Other port numbers are not assigned to any applications, however their range of numbers corresponds to special range or type of applications. Table 2-2 presents some applications according to their transport layer protocol (Layer 4 in both OSI and TCP/IP architecture), as well as the standard port numbers for these applications.

TCP Port Number	TCP Application	UDP Port Number	UDP Application
20	FTP Data	53	DNS
21	FTP Control	67,68	DHCP
22	SSH	69	TFTP
23	Telnet	161	SNMP
25	SMTP	16,384 – 32,767	RTP-VoIP and Video
53	DNS		
80	HTTP		
110	POP3		
443	SSL		

Table 2.2 TCP and UDP port numbers

➢Extended IP ACLs numbers range from 100 to 199 and from 2000 to 2699. All ACLs commands associated with a given number are considered part of the same ACL.

➢Unlike Standard IP ACLs, it is recommended that the location of the interface upon which the Extended ACLs are configured, should be as close as possible to the packet's source, which is the device sending the IP packets to be denied or permitted to reach a specific network resource. Placing the extended ACL near the

packets source allows the discarding of unwanted packets as soon as they access the NAS router interface facing the unsecure internet, which permits the NAS to eliminate routing within the organization inner network, in order to save some bandwidth (Odom, 2009) (Proprofs, 2012).

- The generic syntax of extended ACL's configuration commands is a little different than that of standard IP ACLs', notably because the extended ACLs can match many different fields in the header of the IP packet. Since the possibilities of mix and match of such fields are considerable, there is no specific generic syntax for the extended ACLs configuration that can summarize all possible combinations of theses fields. Thus many versions of extended ACLs syntax exist to reflect such variety of possible combinations.

  Some of the most popular generic syntax associated with extended ACLs are as follows:

- A global command for extended access lists:

  Access-list *extended-ACL-number* **{deny|permit}** *protocol source-IP-address Source-Wildcard destination-IP-address destination-Wildcard* **[log|log-input]** (Odom, 2009)

- Another version of Extended ACL command, with TCP-specific parameters:

  Access-list *extended-ACL-number* **{deny|permit} {tcp|udp}** *source-IP-address Source-Wildcard* [*operator* [*port*]] *destination-IP-address destination-Wildcard* [*operator* [*port*]] **[established] [log]** (Odom, 2009)

Example 2-2

Example of an Extended ACL

In this example, we start by illustrating a network topology, followed by defining the security needs, and the configuration-code of the extended ACL that will help meet such needs. Then, we follow with the explanation of the configuration code, the analysis of the process behind the code and finally, and finally the reflection upon a number of case scenarios that will help the reader grasp the concept behind the working mechanism of extended ACLs. Dynamic ACLs, which are the main focus of this research study, depend fundamentally on Extended ACLs usage.

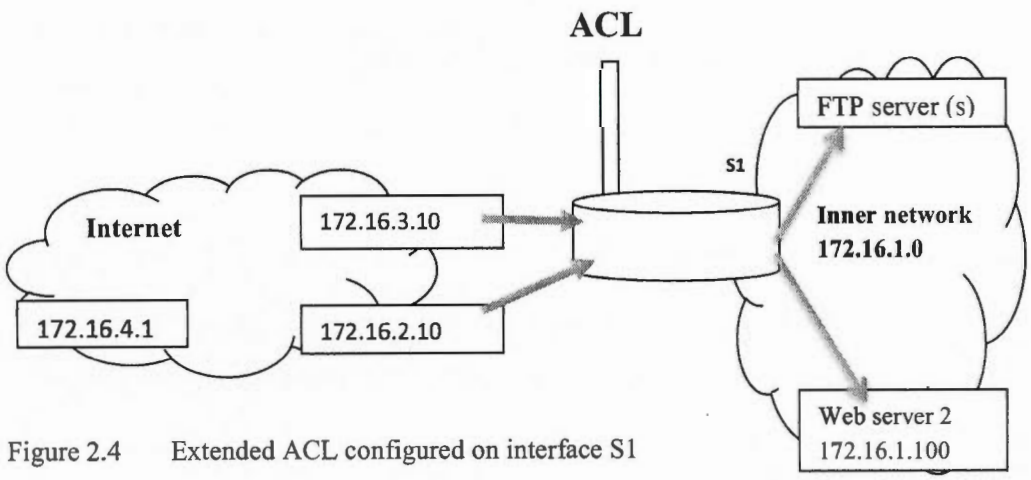

Figure 2.4 Extended ACL configured on interface S1

In this example, and as illustrated by figure 2-4, the organization needs to prevent packets having source IP address 172.16.3.10 from reaching the FTP servers, which are located on subnet 172.16.1.0, while allowing access to packets coming from any other source. At the same time, the organization needs to prevent packets from host 172.16.2.10 to reach inner server 2, the web server at IP address 172.16.1.100, while allowing access to packets coming from any other source.

The configuration will be as illustrated in figure 2-5:

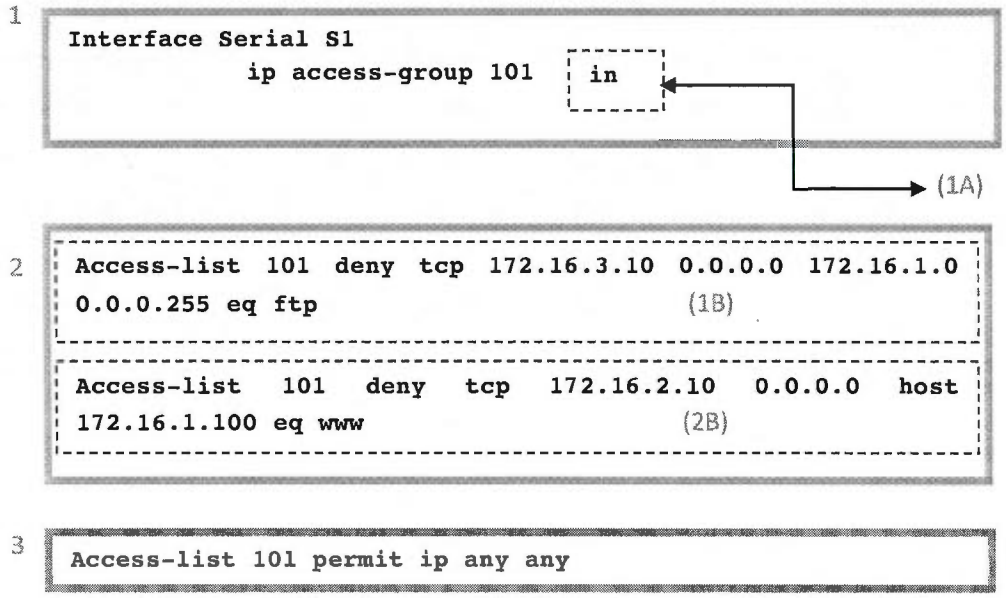

Figure 2.5 Extended ACL configuration code on interface S1

The configuration code, shown in figure 2-5, can be explained as follows:

- In part 1 of the figure, we can notice that this configuration corresponds to applying the access list to the router's interface. The ACL is applied to serial interface (S1), which is the interface closer to the insecure Internet, and therefore the source of the packets to be matched by the ACL.

  The direction of the packets to be examined is specified on the interface using the configuration word "in", in part 1A of the figure. So the extended ACL will match the packets as they go into this router's interface.

- Part 2 of the configuration corresponds to defining the Standard ACL itself, globally on this router, rather than applying it exclusively on a specific interface. The first command, in part 2A of the figure, corresponds to specifying host 172.16.3.10 (using mask 0.0.0.0, to specify that exact IP address), to be denied

access to any FTP server located on the organization's inner network with IP addresses anywhere from 172.16.1.0 to 172.16.1.255. The command ends with "eq ftp" to specify that the connection to the inner network hosts has to be established in order to obtain a Web service.

The second command, mentioned in part 2B of the figure, corresponds to specifying host 172.16.2.10 (using mask 0.0.0.0, to specify that exact IP address), to be denied access to the HTTP server 172.16.1.100 located on the organization's inner network.

- Part 3 of the configuration includes the last globally configured command in this Extended ACL, which is to allow full network access to any other IP packet that doesn't match either of the two ACL statements above.

Now that we examined the extended ACLs' configuration code, let's explain the process that takes place behind it.

- Like in Standard IP ACLs, all matching for the ACL commands' conditions follows the sequential order of these commands. If no match is found, an implicit "deny all" is applied to discard any traffic that doesn't match any of the sequential conditions in the extended ACL.

- The extended ACL's processing starts with application of the ACL to a router's interface so that the filtering process takes effect. We can also notice that the ACL global configuration statements include three conditions (mentioned in part 2 and part 3 of the figure).

- The conditions are matched in a sequential order: Any packets entering the router's interface will be first checked against the conditions in the first command/statement of part 2A in the figure. If a match is made with all the conditions stated in that command, the packet is considered as a match, and no access to the FTP servers'

network will be allowed for that packet, and the packet will be discarded by the interface, as well.

- If no match is found, then the same packet is checked against the conditions in the second ACL statement, in part 2B of the figure. If a match is found, then the packet will be stopped and discarded by the interface.
- If no match is found, then the packet is to be checked against the last condition statement, in part 3 of the figure, which allows access to any other packets coming from the internet.

In this example, if the last command line that permits all other packets is omitted, then all incoming packets will be denied, because of the implicit "deny all", which is a general feature in all kinds of ACLs.

In order to better understand this logic let's consider the following scenarios:

- Let's assume that traffic arrives from host 172.16.3.10 and is destined to any of the FTP servers' IP addresses within the inner network IP address range. The packets of this traffic will match the first condition statement (in part 2A of the figure), and will be immediately discarded at S1 serial interface.

- Let's assume that traffic arrives from host 172.16.2.10 having the HTTP server 172.16.1.100 as a destination. The packets of this traffic will first be tested against the first condition statement (in part 2A of the figure), no match will be found. Then these packets will be tested against the second condition statement (in part 2B of the figure), and this time a match will be found. So the packets will be discarded at S1 serial interface.

- Let's assume that traffic arrives from a third host of the Internet, with IP address 172.16.4.10. Then the packets of this traffic will first be tested against the first condition (in part 2A of the figure), no match will be found. Then these packets will be tested against the second condition (in part 2B of the figure), no match will be found as well. So the packets will be tested against the last condition (in part 3 of the figure),

where a match will be found, and since the condition is to permit access for these packets, they will not be discarded at S1 serial NAS interface.

- Let's assume that traffic arrives from host 172.16.4.10, while part 3 of the configuration is omitted, then the packets of this traffic will first be tested against the first and second condition statements (in parts 2A, and 2B of the figure), consecutively as mentioned in the previous case scenario; and since no match will be found in both cases, and no more condition statements will be there to test against, in the body of the ACL, these packets will be discarded at interface s1, according to the implicit deny all condition statement that characterizes all types of ACLs.

Now that we introduced the extended ACLs mechanisms along with some case scenarios, it will be easier to understand the mechanism of Dynamic ACLs, which are the main subject of the study. In the next section we introduce more types of ACLs since a thorough understanding of these ACLs will lead us for a better understanding of Dynamic ACLs mechanisms, and thus will lead us to reach the goal of this study.

2.4.3 Named ACLs

- Named ACLs function as Standard and Extended ACLs for matching the IP packets. The only difference is that Named ACLs allow the use of names instead of numbers for designating each ACL. Named ACLs are easier to remember and to identify, especially if meaningful names are used.

- In Named ACLs, individual commands lines may be deleted without having to delete the whole ACL with its command lines altogether. This feature provides flexibility in editing /changing the ACLs code, including the modification of the filtering conditions.

  Deleting a whole ACL is a considerable amount of work for the network engineer. Indeed, in order to delete a certain ACL, it must be disabled from all the router interfaces where it has been configured first; then the ACL has to be deleted. And finally, only in case when a configuration change needs to take place in the old ACL, a new ACL has to be reconfigured and enabled on the desired interfaces.

- The IP packet's fields to be matched using a named standard IP ACL are the same as those of a standard IP ACL. Likewise, the IP packet's fields to be matched using a named extended IP ACL are the same as the fields to be matched with a extended IP ACL.

- The Named ACLs' configuration syntax is very similar to the Standard and Extended ACLs' syntax. We conclude this section by giving a couple of quick examples to show the Named ACLs capability to delete single command lines.

Example 2-3

Example of Named ACLs

- In this example, which is the same as the one on Extended ACLs, we will show how a single command of an extended Named ACL can be deleted without changing the configuration of the rest of the Named ACL. If you remember example 2-2 illustrated by figure 2-3, it states that the organization needs to stop any packet from source IP address 172.16.3.10 to reach the FTP servers located on subnet 172.16.1.0, and to allow access coming from any other source. The organization also wants to stop packets coming from source IP address 172.16.2.10 to reach another inner server 2 (a web server), at 172.16.1.100, while allowing the server's access from any other source.

The configuration will be as follows:

```
ip access-list extended riham
deny tcp 172.16.3.10 0.0.0.0 172.16.1.0 0.0.0.255 eq ftp
deny tcp 172.16.2.10 0.0.0.0 host 172.16.1.100 eq www
permit ip any any

Interface Serial 0
      ip access-group riham in
```

You can refer to the explanations about this example in the section entitled "Example of Extended ACL" as these are the very same commands copied in an extended Named ACL which we called "riham". This Extended Named ACL was then applied to the interface as in the previous examples.

Now suppose we changed our mind about denying host 172.16.2.10 access to the web server. We can change the ACL by deleting only the command line that enacts this denial, without interfering with the operation of the ACL or its performance. This can be done as follows:

```
ip access-list extended riham
no deny tcp 172.16.2.10 0.0.0.0 host 172.16.1.100 eq www
```

2.4.4 Numbered ACLs

- Both Standard and Extended ACLs can also be edited using sequential numbers, which is a feature that allows assigning a number to each ACL and to each command line as well. Then, in case one line needs to be deleted, referring to the line by its number is enough to delete the whole line while keeping the rest of the ACL intact. This is much simpler than having to delete a whole ACL, as described above (Odom, 2009).

- Another feature allowed in Numbered ACLs, is the flexibility to add new command statement(s) to the already existing ACL's statements, without having to go through the long process of disabling the ACL off the interface, deleting it, reconfiguring a new ACL including the new statements added, then enabling that new ACL on the desired interface. Numbered ACLs facilitate the manipulation of the ACL statements by providing the possibility of adding new statements, deleting an existing statement or changing it; while keeping the ACL defined and enabled on the router's interface. The use of the numbers to refer to each ACL statement helps specifying the ACL to be deleted or changed. A line number should be specified for a new statement, so that the number falls between two sequential line numbers of any pair of sequential statements already configured in the ACL's body, in order to insert the new statement just at the right place within the ACL. For example if there exist two statements in the ACL, numbered 10 and 20 respectively, then adding a new statement with number 15, inserts that new statement after statement number 10 and before statement number 20.

  Using the same number-referencing technique, statements can easily be deleted off the ACL configuration, without having to deal with the whole ACL.

  Numbered ACLs, thus, facilitate the network engineer's work of dealing with changes to be made to ACLs configuration.

➢Numbered ACLs have other useful features that add flexibility to the network configuration process, however since they are not the main focus of this study; we will skip such unnecessary details.

Example 2-4

Numbered ACLs Example

This example shows how using numbers to refer to the standard ACL statements transforms the ACL into a Standard Numbered ACL, and how we can delete or change the configuration of a single existing command statement within the ACL, just be referring to that statement's specific number, without changing the configuration of the rest of the Numbered ACL, and without the need to disable, delete it, rebuild it and re-enable it again on the interface upon which it is configured.

Thus suppose that the already existing ACL configuration code is as follows:

```
ip access-list standard 20                    (1)
    deny 172.16.3.10
    deny 172.16.2.10
    permit 172.16.1.0 0.0.0.255
```

This configuration doesn't show the line numbers identifying each statement within that ACL, which makes it hard to refer to individual statements. In order to retrieve the original router configuration for the previous code while showing the numbers corresponding to each statement of this ACL, we use the following command:

```
do show ip access-list 20                     (2)
```

The result of command (2) will show the retrieved ACL, which is the same ACL after including the numbers corresponding to each statement:

```
Standard IP access list 20                    (3)
    10 deny 172.16.3.10, wildcard bits 0.0.0.0
    20 deny 172.16.2.10, wildcard bits 0.0.0.0
    30 permit 172.16.1.0, wildcard bits 0.0.0.255
```

Now that we can see the numbered statements, we can change the ACL's statements according to our needs. Specifying the statements to be changed will be done by referring to them using their numbers. For example in order to add a new statement to be placed between line 10 and 20, we give it any number that falls within that range, say number 15, as shown in command (4). We can also delete statement number 20, as shown in command (5).

```
15 deny 10.1.1.1                                    (4)
no 20                                               (5)
```

In order to retrieve the ACL's configuration after the changes have taken place, we use the following command:

```
do show ip access-list 20                           (6)
```

The retrieved ACL's code after the changes is:

```
Standard IP access list 20                          (7)
    10 deny 172.16.3.10, wildcard bits 0.0.0.0
    15 deny 10.1.1.1, wildcard bits 0.0.0.0
    30 permit 172.16.1.0, wildcard bits 0.0.0.255
```

Part (1) corresponds to configuring the ACL itself using a number "20", in order to refer to that ACL including all its statements, as same as we saw in the previous types of ACLs. ACL 20 as we see, contains 3 ACL's statements.

In part (2), the command is to show the configuration code of the ACL. Since the configured ACL is a standard Numbered ACL, the router shows the numbers corresponding to the ACL. It also assigns some sequential numbers to the ACL command lines.

These sequential numbers have a gap, to allow the possibility of adding other commands in between if needed in the future. Here, we added the command line 15 to deny access to host 10.1.1.1. Choosing the number 15 is to inform the router to place that configured statement between statement 10 and before statement 20. Also, we can delete an ACL statement just by referring to its number, as we deleted statement 20, as shown in command (5) of the example. Thus Numbered ACLs

provide a very practical means that can help the network engineer to easily change the different ACLs configurations.

2.4.5 Reflexive ACLs

- Reflexive Access Control Lists (ACLs), known as IP-Session-Filtering security, are another type of ACLs used by an organization to prevent a class of security attacks for each allowed TCP or UDP session, on a per session basis.

- Reflexive ACLs react to the first incoming packet of a new connection session, recording the Source IP address along with the UDP or TCP port, on one hand, and pairing them with the organization resource IP address and port (in other words: the destination) on the other hand. This technique helps filtering all incoming traffic except packets sent by that specific Source IP address paired with that specific session.

  The main benefit of Reflexive ACLs is to allow a better control of network access security, by enabling the router to filter IP packets dynamically, based on the connection sessions established with the organization resources.

  How do Reflexive ACLs work?

  Reflexive ACLs are usually nested within the extended named ACLs that are applied to the interface of a border router (firewall router), enabling it to filter outbound traffic. Once a connection session starts, the router adds entries to the temporary ACLs, allowing the inbound reply packets only for the duration of the connection session. The Reflexive ACL entries are: the source and Destination IP address and the TCP/UDP port number.

2.4.6 Time-based ACLs

- Time-based ACLs are similar to Extended ACLs in function, however they also allow the addition of time-constraints to the configuration of either numbered or named ACLs, so the latter would permit packet filtering (according to the ACL's conditions) during a specific period of time, e.g. during one day or specific days of the week (Odom, 2009).

How do Time-based ACLs work?

The Network Operating System adds or removes different statements of the ACL during the appropriate time of the day in order to control the access to certain resources of the organization during that specific period of time.

Example 2-5

Time-based ACLs Example

In this example, the organization needs to allow FTP access to the hosts of an outer network 172.16.3.0/24 only on Mondays, Wednesdays and Fridays, during working hours.

So the configuration is as follows:

```
Interface Serial 0
    ip access-group 101 in

Access-list 101 permit tcp 172.16.3.0 0.0.0.255 172.16.1.0
0.0.0.255 eq ftp time-range EVERYOTHRDAY

Time-range EVERYOTHERDAY
Periodic Monday Wednesday Friday 8:00 to 5:00
```

In this example, the time range called EVERYOTHERDAY is the time constraint instructed to the ACL to respect while permitting the hosts from the 172.16.3.0/24 subnet to access the FTP servers.

Dynamic ACLs are another type of ACLs that filter unwanted packets at the organization's border. Dynamic ACLs are the main focus of this study and will be thoroughly explained in the following Chapter, Chapter 3- Dynamic ACLs.

In general, chapter 2 can be considered as an explanation of the IP ACLs filtering mechanism, introducing a number of IP ACLs types, along with a thorough explanation of their usage, their mechanism and their configuration. The chapter's importance lies in its nature as a corner stone necessary for the structure of this study, explaining the typical organization's topology where IP ACLs are implemented in general, their filtering needs and their variety according to the organization's security needs. Thus the chapter is a crucial base upon which the following chapters are built, especially that it introduces a list of IP ACLs categories that ends with Dynamic ACLs, which are covered in details in the following chapter and which are the main focus of the study. The chapter provides information based upon a deep analysis of the theoretical information provided through the different references used, as well as through detailed developed configuration examples, from which the reader can better understand the theoretical explanation of the different concepts and make a sensible evaluation for the different potential implementation choices.

In details, the chapter starts by a description of ACLs, their filtering process, and their configuration. Then the chapter follows by listing a number of IP ACLs categories, as well as their application necessities and their configuration stated in the form of explanatory example that are developed specifically to support this research study. Thus the chapter presented Standard ACLs, Extended ACLs, Named ACLs, Numbered ACLs, Reflexive ACLs, and Time-based ACLs, while mentioning Dynamic ACLs as a final category that belongs in the list, and that will be the main focus of the study, requiring a whole chapter in order to be explained in greater details, allowing its in-depth analysis.

CHAPTER III

DYNAMIC ACLS

Remote user authentication and remote user access restriction for specific inner resources can be obtained by applying one of two methods:

1. Dynamic ACLs, which are the main focus of our study.
2. Auth-proxy[11], which will be discussed in chapter 5, entitled "Authentication and dynamic ACLs".

Thus authentication proxies are one filtering technique that can be comparable to Dynamic ACLs in terms of security mechanism.

In this section, we explain Dynamic ACLs necessity, mechanism, configuration and authentication.

3.1 Dynamic ACLs Purpose

Dynamic Access Control Lists (Dynamic ACLs), known as Lock-and-Key security, are a special kind of ACLs used when an organization wants to grant secure IP access to its servers and inner resources to a small set of users that have to connect remotely, using their home PCs or their laptops in order to reach these resources.

These remote users, usually keep changing their IP addresses as they lease new IP addresses using the Dynamic Host Configuration Protocol (DHCP) every time they establish an Internet connection to start a session with the organization's network.

Thus controlling the user secure access by limiting the range of the source IP addresses connecting to the organization will be impossible, since the users keep

[11] Auth-proxy, though might include an Authentication server like the ones used with Dynamic ACLs, works differently as a concept.

changing their IP addresses into unpredictable ranges due to the DHCP dynamic address acquisition process.

Therefore Dynamic ACLs implementation will help the organization enforcing secure user access by authenticating the different users, regardless of how unpredictable and dynamically changing the source IP addresses they use. In order to perform the filtration role of these user connections, Dynamic ACLs can be used as substitutions for many other filtering/controlled connection methods, like Virtual Private Networks (VPN)[12], Traditional ACLs and Proxy servers (Authentication servers).

Before we expand the details behind the configuration of Dynamic ACLs, let's compare them to a number of other filtering techniques, which will help us better explain these ACLs and how they work.

Let's first consider the use of VPNs versus Dynamic ACLs. When the remote users need to access the organization's inners network through a public network (like the Internet), usually organizations allow the establishment of such a connection through the use of VPNs, which provides a great level of security for transmitted data. However, once such a connection is established, security control of such connection allows remote users to freely access all the organization's inner resources risking to compromise all them all, as if that access is locally done from within the organization's network. This restriction done by the organization cannot be considered a full security control, since the organization cannot restrict access to some of its inner resources, while using a VPN connection, except with the use of a regular ACL combined with the VPN connection (E-Tutorials Lock-and-Key Configuration, 2012). This ACL only filters packets based on layer 3 and layer 4 conditions, as explained in chapter 2. Thus they don't go all the way to filter packets according to upper layers concerned with session establishment, data representation and application involved in the connection.

[12] VPNs are Virtual Private Networks, networks that encrypt all traffic while it flows over the unsecure Internet.

Another drawback of using a regular ACL resides in the fact that restricting traffic cannot be personalized according to each user's needs and allowed accessibility. As the ACLs only authenticate the remote device (or the traffic it belongs to), the remote user is not authenticated nor is his access to the organization resources controlled on a per user basis. In fact this ACL will grant the same access rules for all users to reach the same specific organization inner resources[13]. This is the not the right answer to meet the need of the organization to personalize remote access restriction according to its security rules and according to each user's needs.

Since traditional ACLs configured on the organization's router (border devices) filter users based on their IP addresses, this filtering cannot be applicable in cases where the IP addresses keep changing dynamically. It is impossible to edit every single ACL configured on the organization resources in order to cope with the dynamically changing user-IP addresses. Also, the maintenance of the integrity of these ACLs to reflect such changes is impractical.

Another problem that might happen if we try to use traditional ACLs to secure dynamically changing user IP addresses is the possibility of creating security holes by constantly changing the current ACLs according to the users ever changing IP addresses. In case of using traditional ACLs, there will be two types of challenges for the organization to face: challenges when keeping track of the user IP address changes, as well as challenges when trying to adapt existing ACLs to cope with these dynamic changes, as explained in the chapter 2. The security risk of creating security holes is presented when forbidden users get hold and use old permitted IP addresses, not currently used by permitted user hosts, since keeping track of the permitted versus forbidden IP addresses will represent a great challenge to the organization in this case.

The use of dynamic configuration of ACLs, in the lock-and-Key ACL, is similar to their use in Reflexive ACLs and Context-Based ACLs. What differs is

[13] The remote user authentication will be described in details in chapter 4, entitled "User Authentication". The Personalization of the remote user access to the organization's resources topic will be analyzed in chapter 5, entitled "Authentication and Dynamic ACLs".

what triggers the addition of the dynamic entries. Reflexive ACLs and Context-Based ACLs add dynamic entries into the ACL configuration based on their acceptance of the inspected traffic, and then they allow only the traffic corresponding to the original remote user's device that initiated the connection through the ACL to return back into the network. Lock-and-Key ACLs, on the other hand, have to authenticate the remote user first, then, depending on that authentication, specific dynamic entries are temporarily added into the ACL configuration, already applied to the router's interface. So lock-and-Key ACLs permit access to the user for organization resources that normally would be denied to him. Usually, lock-and-Key ACLs are combined with other types of ACLs, especially extended ACLs, as mentioned in chapter 2.

Extended ACLs, as explained before, contains many static conditions. Once configured and enabled on a router's interface, they remain active until disabled, filtering all packets based on the same conditions. Extended ACLs don't allow dynamic entries to be added to the ACL's body, in order to suit the permissions granted to every user. Thus, they don't allow controlling this access, neither do they allow user authentication mechanisms.

According to Orbit Computer Solutions (Orbit Computer Solutions, 2012), "Some of the many security benefits of Dynamic ACLs over standard and static extended ACLs are:

1. The use of an authentication mechanism for individual users.
2. Reduction of the opportunity for network break-ins by network hackers.
3. In many cases, reduction of the amount of router processing that is required for ACLs.
4. Simplified management in large internetworks.
5. Creation of dynamic user access through a firewall, without compromising other configured security restrictions."

3.2 Dynamic ACLs usage

Dynamic ACLs are used to permit secure layer 3-IP packet access for remote user connections made to the organization's resources.

Though many physical connection might be secured by Dynamic ACLs including DSL and cable, where the user acquires his own IP address from an Internet Service provider, such methods will still represent a single remote IP address (representing one user or a small number of users) trying to establish a connection to the organization's inner resources through Dynamic ACLs. Other broadband, multipoint connections (where many users try to connect to the organization's resources), are not to be secured through the use of Dynamic ACLs. This limited usage is due to the fact that a given Dynamic ACL involves VTY connections (usually Telnet), which only allow 16 simultaneous user authentication processes to take place over a single router interface protected by a Dynamic ACL.

Thus, according to E-Tutorials (E-Tutorials - Lock-and-Key Overview, 2012), Dynamic ACLs are needed in the following situations:

1. Allowing inner network temporary access to certain remote user(s) connecting through the Internet.
2. Allowing certain inner hosts to access a specific remote host/network.

The study will focus on the first implementation situation of Dynamic ACLs.

3.3 Dynamic ACLs mechanism

Dynamic ACLs may be considered as Custom made ACLs, designed for every user host's IP address. Actually, Dynamic ACLs start configuring themselves automatically and dynamically depending on the dynamic users IP addresses.

Triggering this dynamic configuration of the ACLs depends on the very first step, which is the User authentication step. Without this authentication step, the ACL itself cannot be triggered or built, and no access is granted to that user's host IP address.

Once the user passes the first step and is considered authenticated, the router (or the organization edge device) dynamically reconfigures the ACL, permitting access to the host used by that authenticated user.

Actually, the router then will dynamically reconfigure the ACL using that host IP address temporarily, while permitting that user's access towards the inner network mentioned in the ACL itself. Then after the user session is done, or after a certain period of inactivity, depending on the dynamic ACL configuration, the router removes the temporary IP address from the ACL and access is restored to its previous status.

Triggering a Dynamic ACL starts by a connection made from the remote user to the organization's network. Usually this connection is a Telnet connection. Telnet is a network protocol that provides bidirectional interactive text-oriented communication with a remote host command line interface (CLI) using a virtual terminal connection (VTY). Telnet is one of the first Internet standards and was documented in RFC 15, extended by RFC 854 (Wikipedia - Telnet, 2012).

Telnet belongs to layer 7 (application layer) within the OSI model stack, and can always be replaced by other connection methods, as we will discuss in chapter 5, entitled "Authentication and Dynamic ACLs".

According to Orbit Computer Solutions (Orbit Computer Solutions, 2012), "This type of access control list is basically reliant on Telnet connectivity, authentication and extended ACLs. Lock-and-key is configured using IP dynamic extended access lists. This can be used in conjunction with other standard access lists and static extended access lists. "

The Dynamic ACL process is usually combined with the use of Telnet as an access method for the remote user to reach the network resources. However, in order to guarantee its security aspect, this process is quite complicated since it depends on the success or the user connectivity through Telnet as a first step that allows the dynamic ACL mechanism to take place. Though Telnet or any other VTY (line) access technique is considered independent from the Dynamic ACL mechanism, since the Dynamic ACL process is

triggered from within the line connection, it is essential that the connection takes place successfully.

Thus the whole process includes two types of authentication that will take place:

- An authentication allowing the VTY connection only (Telnet).
- An authentication allowing for reaching the network resources (Through dynamic ACLs).

  This second authentication takes place within the Telnet connection configuration, once the user is authenticated for the use of Telnet and is getting authenticated to access the organization inner resource(s). Thus this authentication is referred to as "the user authentication process" and it is the main focus of the study.

Both authentications are independent from each other security wise, however Dynamic ACL authentication process is dependent on the line connection authentication connectivity wise.

Each authentication has an individual timeout, thus we end up having two types of timeout involved in the Dynamic ACLs mechanism, as follows:

1. The first authentication's timeout "the Idle Timeout":
   An optional timeout parameter can be configured to limit the user access through the Telnet connection. This timeout is specified within the VTY's autocommand that enables the Lock-and-Key user authentication. If the Idle Timeout is not specified, the default is to never timeout the dynamic ACL entry.

2. The second authentication's timeout "the Absolute Timeout":
   Another optional timeout parameter can be configured to limit the user access to the inner resources through the Dynamic ACL entry. This timeout is specified within the extended ACL, in order to define an absolute-access duration for the dynamic ACL entry that will take place within the extended

ACL, once the remote user is successfully authenticated. If this timeout is not specified, the default is to never timeout the dynamic ACL entry within the extended ACL.

Thus it is recommended to specify either types of timeout within the Lock-and-Key configuration; otherwise the remote user access will remain active indefinitely to the organization's inner resources.

Either timeout parameter (idle or absolute) can range from 1 to 9999 minutes.[14] And usually the absolute timeout value set is greater than the idle timeout's.

A dynamic ACL is removed from the NAS configuration when the router reboots, when its entry in the extended ACL is manually cleared, when the idle timeout is reached, or when the absolute timeout is reached. Even when a router's configuration is saved, the dynamic ACL entry created once a user authenticates to the NAS is not saved. However once the router reboots, that dynamic entry can be easily recreated again once the user re-authenticates.

Though the Dynamic ACL's filtering mechanism takes place at layer 3 of the OSI model, as well as all types of IP ACLs as we explained in chapter 2, the user authentication process takes place at layer 7 as the user starts a layer 7-Telnet session to connect to the organization's network.

Dynamic ACLs mechanism can be clarified by the processes shown in figure 3-1. These processes will help justifying the mechanism steps that will follow it.

[14] More detailed explanation about the timeout parameter will be given in the following sections.

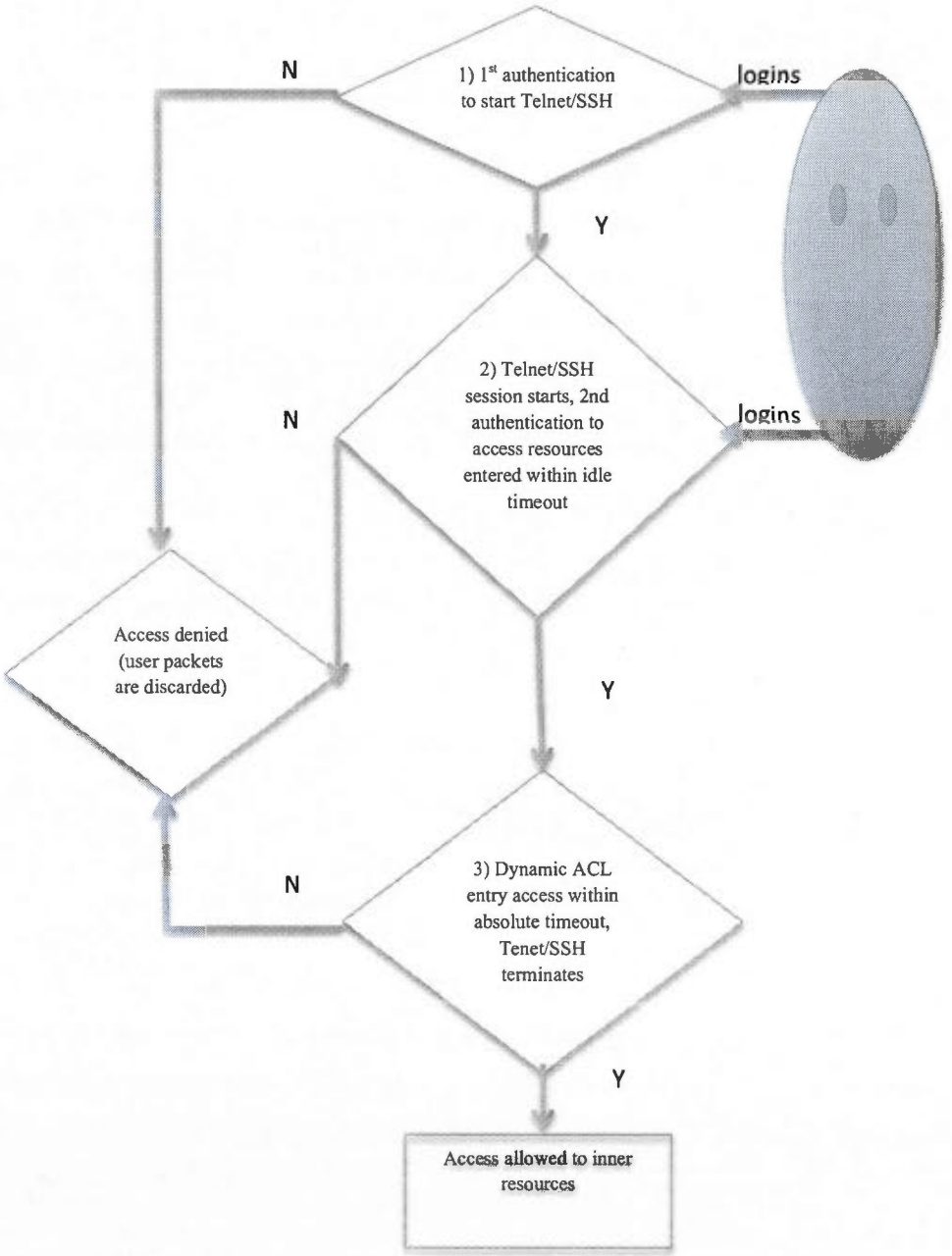

Figure 3.1 Dynamic ACLs mechanism Processes

Dynamic ACLs mechanism is performed according to the following steps:

1. Telnet to the Network Access Server (NAS):

    1.1. The user is blocked from accessing the NAS (the border r outer configured with the Dynamic ACL) except if he uses a Telnet connection[15]. Thus the existing ACL must allow Telnet (or SSH) so that the user can enter his credentials to open a Telnet session, and then connect via a virtual Terminal port to the router (Telnet/SSH port).

    1.2. The router receives the Telnet credentials packet (1st authentication), opens a Telnet session; and within that session, the router prompts the user for his credentials to authenticate him (2nd authentication).

    1.3. The user enters his credentials according to the authentication method defined for the Telnet access method, on the NAS router.

2. The NAS authenticates the user

    2.1. The NAS router authenticates the user (2nd authentication) locally or through a remote authentication process; according to the authentication method defined within the VTY connection (Telnet)[16].

---

[15] In order to start a Telnet connection, the user has to enter his credentials and has to be allowed to start that connection through the configured ACL.

[16] The VTY connection configuration contains an autocommand that will trigger the Dynamic ACL entry, once a successful authentication process takes place.

3. The NAS router adds the Dynamic entry related to the user

3.1. If the user is authenticated, then:

a. The Telnet session is dropped (since the Telnet/SSH connection is only needed to perform the authentication step, and it is no longer needed).

b. One dynamic ACL entry is added to the extended existing ACL: so the inbound interface is configured to allow the user temporary access to the inner network resources, which would be otherwise denied through the regular static extended ACL.

c. The user exchanges data, through the firewall/NAS router, with the server/inner resources he/she needs to reach.

d. When an absolute timeout is reached, the Dynamic ACL entry is taken off the original extended ACL, and the inbound interface is reconfigured back to its original state to block all access attempts.

If the temporary entry is not used before the idle timeout (configured within the Telnet session's autocommand) is up, the telnet session closes and the temporary entry will expire.

If neither an absolute timeout nor an idle timeout is configured, the dynamic entry (permitting access to that specific user) will remain configured indefinitely on the interface until it gets manually removed by a network administrator. This represents then another way to end up that temporary access permission.

3.2. If the user is not authenticated then:

a. The user has no access to the inner network resources, except for the resources specified inside the original static extended ACL.

b. The user is prompted once again, for the authentication information to be entered.

3.4 Dynamic ACLs Configuration (using local authentication)

Dynamic ACLs are configured using IP dynamic extended Access lists, and have been introduced in Cisco IOS release 11.1.

The three main steps for configuring Dynamic ACLs are as follows:

1. Creating extended ACL, permitting either Telnet or SSH as a remote access method into the NAS, and setting up a placeholder in the extended ACL for the dynamic ACL entry that will be generated dynamically once the user authentication process successfully takes place. That placeholder will be a reference to the Dynamic ACL within the extended ACL, thus the Dynamic ACL configured name must be unique within the NAS.

2. Defining the authentication method for remote users: whether it is local, external through the use of an authentication server, or using VTY (line) password.[17]
3. Enabling the authentication method on the NAS VTY line so that the NAS will be able to create a Dynamic ACLs entry within the extended ACL that references it, and that is configured on the same interface.
   (E-Tutorials - Lock-and-Key Overview, 2012)

[17] Further explanation of the different methods of user authentication within Dynamic ACLs will be found in chapter 5, entitled "Authentication and Dynamic ACLs".

The extended ACL should be created on the NAS external interface and should permit Telnet or SSH access to the IP address of the NAS external interface. The lock and Key entry that is embedded within that extended ACL defines which inner resources the user will be allowed to access (E-Tutorials - Lock-and-Key Overview, 2012).

Dynamic ACLs configuration cannot be explained without explaining the corresponding operation processes that take place as a result of the configuration process. Thus, let's explain the sequence of Dynamic ACLs operation processes in this section.

Dynamic ACLs operation starts with an extended ACL that blocks all traffic to the NAS except for Telnet authenticated traffic (1st authentication). The Telnet connection is shutdown after a successful user authentication (2nd authentication), and a single entry dynamic ACL is added to the extended ACL, permitting temporary access to a specified user into the inner network.

Because Dynamic ACLs' were configured in a particular way, the corresponding operation steps take place in a particular sequence and logic. In this section we will analyze the general operation steps that will help understand how Dynamic ACLs processes take place, as well as to guide us through future Dynamic ACLs configurations. The steps are as follows:

1. Assuming that, on the NAS, the user authentication method is defined, including a username and password (local authentication)

2. Assuming, that on the NAS, the ACL is defined as follows:
   Allow Telnet session establishment between any and all source IP address and the NAS IP address as a destination.
   This Telnet session is defined as follows:

   a) Check if the user login (username/password) is as defined in step 1. (authenticate the user)

b) Once the user is authenticated,

b).i. Terminate the Telnet session.

b).ii. Make one temporary dynamic entry in the extended ACL, specifying that permitted user IP address as source address.

b).iii. Execute the autocommand that is defined within the ACL.

b).iii.1. Allow the user specified in the dynamic ACL (that's the source IP address configured dynamically) to access the inner network resources (that could be the IP address of the inner NAS port).

b).iii.2. If a shutdown timer (idle timeout) is specified to allow a certain period of time for the Telnet access, and if that idle timeout is up before the user gets authenticated, the Telnet session closes and the user has to re-authenticate.

If the user uses the Telnet connection and gets authenticated before the idle timeout is up, then the absolute timer takes effect and the dynamic access session closes with respect to the absolute timer.

A few points need to be clarified about the steps mentioned above.
These points are as follows:

1. The user authentication can be either local (on the firewall router/NAS) or remote (through external AAA server)[18]. This topic will be discussed in more details in the following section.

[18] Line password is another authentication option that can theoretically be used by dynamic ACLs, however it is not used in real world environments, since all users would be using the same password.

2. The dynamic entry is, as mentioned before, personalized according to each user's IP address. Thus each user will be either granted or denied access according to the organization's policies related to that user. However, the Lock-and-Key configuration itself, when performed on a specific router's interface, doesn't allow a personalized access policy to each user in respect to the destination servers (inner network premises) allowed to that specific user (according to his user privileges), excluding other disallowed premises[19]. This is because Dynamic ACLs can only have one dynamic entry that must encompass all external users accessing the organization's resources. Thus, once authenticated, all users will have a dynamic entry ACL to allow them to access the same destinations (same premises) as configured in the Lock-and-Key ACL, even though they're using different source IP addresses to access the very same destination (organization's inner resources). This is why Dynamic ACLs are usually used to securely connect a small number of remote users to an organization's bastion server, where they are configured. The bastion server is a network device that is designed and configured to withstand attacks. NAS routers and firewalls can be considered as bastion hosts due to their exposure to security attacks (Wikipedia - Bastion Host, 2012). Once connected to the bastion server, the users will then be able to access other organization inner resources (E-Tutorials Lock-and-Key Configuration, 2012). In the study, we will focus on Dynamic ACLs implementation on the NAS router to directly access the organization's inner resources.

3. The temporary access time period can be defined either as an idle timeout within the VTY configuration code, or as an absolute timeout. By the end of either timeouts, the temporary ACL entry, that will permit the limited time user access, will automatically disappear off the configuration, through the autocommand operation. However, if none of the timeouts is configured, the

[19] The personalized user access policy will be discussed in details in chapter 5, entitled "Authentication and Dynamic ACLs".

remote user access period will be indefinite, as the temporary ACL entry will stay forever within the configuration, without being taken care of by the autocommand. In that case, the only way to take the temporary ACL entry off the configuration, will be by the network administrator, who will manually clears the Dynamic ACL entry off the configuration.

The generic syntax for the dynamic entry of the Lock-and-key configuration is as follows:

A global command for extended number Access lists (Cisco Configuring IP Access lists, 2007):

Access-list *extended-ACL-number* ***dynamic*** *name {**deny**| **permit**} [protocol] {source-IP-address Source-Wildcard*|**any**} *{destination-IP-address destination-Wildcard*|**any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**|**login-put**] [*operator destination-port*|*destination- port*]

The extended ACL has to be enabled on a given interface, so that it becomes functional. The interface on which it should be configured is the NAS external interface facing the internet.

We can specify the VTY (line) connection to configure Telnet, as follows:

Line vty *line-range*
Login local
(Odom, 2009)

Here, we mentioned the range of Telnet connecting lines as well as the authentication method that will be performed as a result of such a connection. The authentication method used is local authentication, performed by this router. Other forms of authentication will be mentioned later in this chapter.

In order to better understand this syntax let's analyze an example of Dynamic ACLs.

3.5 Example of Dynamic ACLs using local authentication

In this section, we will demonstrate the configuration followed by a thorough explanation of the configuration code, as well as an analysis of the dynamic ACLs mechanism.

In the example, the organization needs to stop any access from reaching its resources, except through the use of Telnet and local authentication (authentication occurring locally on this NAS/firewall router). Let's assume that a user using IP address 172.18.3.10 needs to login remotely to the organization resources (host 10.1.1.1), represented by the network 10.1.1.0 with network mask 255.255.255.0.

This user will initially only be allowed to login using Telnet, in order to authenticate to the organization's resources. Then provided the authentication is successful, the user will be allowed complete access through the Lock-and-Key ACL.

The configuration will look like the following:

1
```
Interface Ethernet 0
  ip address 172.18.3.3 255.255.255.0
  ip access-group 101 in
```

2
```
Access-list 101 permit tcp any host 172.18.3.3 eq telnet(1)

Access-list 101 dynamic mylist timeout 100 permit ip
any   10.1.1.0 0.0.0.255                                (2)
```

3
```
Line vty 0
  login local                                           (1)
  autocommand access-enable timeout 4                   (2)
```

Figure 3.2 Dynamic ACLs configuration using local authentication

Explanation of the Dynamic ACLs' example configuration:

Part # 1 of the configuration corresponds to applying the Lock-and-key access list to the router's interface. The Lock-and-key ACL uses an Extended ACL; therefore, as previously explained, it will be applied to the router's interface facing the outside network, near the source packet to be examined. This interface is an Ethernet E0, having the IP address: 172.18.3.3.
The direction of the packets to be examined is "in", so the extended ACL will match the packets as they go into this router's interface, from the outside network.

Part # 2 of the configuration corresponds to defining the Extended ACL itself, globally on this router, rather than applying it exclusively on a specific interface. The first command/statement # (1) permits any host to get a layer 4-connection type using TCP only, and application connection (layer7) using Telnet only to the destination 172.18.3.3, which is the telnet port of the NAS/firewall router.

Command/statement # (2) in this Extended ACL is ignored until the lock-and-key mechanism is triggered. It corresponds to allowing the authenticated user referred by the key word "any", to dynamically access all inner resources on network 10.1.1.0 255.255.255.0, for a limited time of 100 minutes (absolute timeout).

Part # 3 of the configuration corresponds to the VTY line configuration (Telnet configuration). In command # (1), the type of the authentication method to be used is specified: the authenticated is to be performed locally, directly by the router.

Command # (2) in part 3 of the configuration, corresponds to an autocommand that is configured to trigger the dynamic ACL entry, creating it within the original extended ACL in part 2. That entry can be triggered only during an

active Telnet session where the user authentication has successfully taken place within a 4-minute period of time (idle timeout).

The operation mechanism of the Dynamic ACLs' example configuration:

The mechanism behind Dynamic ACLs consists of two distinct authentication processes that the user has to successfully go through in order to reach the inner organization resources. The first authentication process is to enable the user to start a Telnet connection with the NAS router. The second authentication process is to enable the user to start an IP connection to the organization's inner resources.

The mechanism will allow the following scenario to take place: once the user successfully enters his credentials during the first authentication process (Telnet, specified in part # 3), a Telnet connection is established with the NAS router. Next, the user is prompted to enter his credentials for the second authentication process where the router attempts to authenticate that user locally (part # 3, command (1)). If the second authentication is successful, the autocommand (in part # 3 command (2)) executes, triggering a new generated "dynamic ACL entry" to be added into part # 2 of the configuration, and the Telnet session terminates. The authenticated user is then allowed access to the organization resources through the generated dynamic ACL entry. Throughout our study, we will focus on the second authentication process, while discussing the security aspects of the Telnet connection (the first authentication process) as well[20].

[20] The topic of Authentication will be elaborated in details later in this chapter, as well as in chapter 4 and 5.

In this section we focus on the mechanism of each part of the configuration code in more details:

When looking at part 2 of the configuration, we can notice that, like in standard and extended IP ACLs, all matching for the conditions stated in the ACL commands is made in a sequential order of these commands; and if still no match is found with the examined IP packet, an implicit deny all is applied at the end of the ACL's body (after statement (2) in part # 2 of the ACL configuration) to discard that packet.

Tracing that ACL statement, any packet entering the router's interface will be first checked for all the conditions in the first command/statement (1), which states a layer 4 connection type using TCP only, and application connection (layer7) using Telnet only to the destination 172.18.3.3, which is the telnet port of the NAS/firewall router If a match is obtained with all the conditions stated in that command, then the packet is considered as a match, and a Telnet access to the routers' network will be allowed for that packet, opening a Telnet session for the user. If a packet doesn't match any of the conditions stated in the first command exactly, then no access is allowed and the packet is discarded by the interface.

Since the second ACL statement (statement #2) of part # 2, will only be processed after the Telnet connection takes place, we will only proceed with explaining its operation after explaining the Telnet connection in part # 3, which corresponds to the Telnet session itself including the autocommand that will trigger the Dynamic ACL entry itself, into part 2. Giving the precedence to part # 3 explanation is important to understand the occurrence of the authentication process in the same sequence that actually happens during the ACLs filtering process.

In part # 3, the autocommand is triggered only once the user authentication (2^{nd} authentication) using the router succeeds, allowing the user access to the organization's inner resources. The autocommand creates the temporary ACL entry within part 2 of the configuration, which corresponds to the Dynamic entry "mylist".

The autocommand configuration states that the access will be enabled for a user who's been authenticated locally through NAS router (2^{nd} authentication). However, as shown in part # 3, statement (2), that user has a window of only 4

minutes, once he becomes authenticated[21] (1st authentication) and logged in through Telnet, to make use of the Telnet connection in order to authenticate directly to the NAS router itself[22]. If during that 4-minute session, there is no activity by the user, the Telnet session connection closes and the user has to authenticate again.

This 4-minute timeout corresponds to the idle timeout for the user once he authenticates to use Telnet. However, if the user actually uses the Telnet connection to attempt to authenticate locally in order to access the organization inner resources, within a 4 minute-window of time that starts at the moment he authenticates for the Telnet connection, the Telnet session will still terminate as well. However, in case the user authentication to reach the resources is successful, closing the Telnet session will only take place after triggering the autocommand, which will generate a temporary inbound access list entry on the router's interface, allowing access for the authenticated user through the firewall/router.

Thus the idle timeout starts with the user authentication to use Telnet (the first authentication process), and terminates either by exceeding that timeout period or by successfully authenticating to access the inner resources through the temporary ACL (the second authentication process).

The maximum period of time for that temporary access through the ACL's opening is configured in the absolute timeout, which takes effect once the Telnet session terminates while the authenticated user is logged into the router.

Once the absolute time is reached, the connection session permitted by the temporary dynamic ACL will close, and the dynamic entry is taken off the extended original ACL. The absolute timeout is mentioned in the extended ACL's body, in the command (2) of part # 2 of the configuration (timeout 100). If the user still needs extra connection time more that the 100 minute-absolute timeout, he has to redo the whole process from scratch (using a Telnet connection to authenticate).

---

[21] The user will be authenticated here only in order to open up a Telnet session with the NAS router, while not being allowed to access any other inner organization resources, yet.

[22] The user will be authenticated here only in order to be allowed access to the inner organization resources.

Coming back to the operation corresponding to the body of the Extended ACL, since the autocommand is triggered, a temporary ACL is entered as a complementary command for the command (2) of part # 2. The triggered dynamic entry will specify one or more IP address to be allowed access into the network specified in command (2) of part # 2 of the configuration. After the autocommand is triggered, the configuration of the Lock-an-Key ACL, is, in fact modified, so that the generated dynamic entry looks like figure 3-3:

1
```
Interface Ethernet 0
  ip address 172.18.3.3 255.255.255.0
  ip access-group 101 in
```

2
```
Access-list 101 permit host 172.18.3.10 10.1.1.0 0.0.0.255
```

3
```
Access-list 101 permit tcp any host 172.18.3.3 eq telnet (1)
Access-list 101 dynamic mylist timeout 100 permit ip
any 10.1.1.0 0.0.0.255                                   (2)
```

4
```
Line vty 0
  login local                                  (1)
  autocommand access-enable timeout 4          (2)
```

Figure 3.3 Dynamic ACLs configuration including the Dynamic entry

According to figure 3-3, the new command in part # 2 of the ACL's body corresponds to the dynamic entry specifying the access permission for packets coming from the authenticated IP address 172.18.3.10 into the firewall/router outer interface 172.18.3.3. This command line is only added to the Lock-and-Key configuration after the user authenticates locally (the second authentication process) using the NAS router while using a Telnet connection and after the dynamic ACL entry is triggered through the autocommand (command (2) of part # 3). Triggering the dynamic entry is to allow access to one or more source IP addresses mentioned in the variable "mylist". This variable contains the results provided by the newly

created dynamic entry (which allows access to the authenticated user 172.18.3.10), as we will see in figure 3-3.

Then the Telnet connection drops so that the user can have his own ACL entry added to the configuration to allow him access to the organization's inner network 10.1.1.0.

3.6 Dynamic ACLs authentication

Now that we introduced an example of Lock-and-Key ACLs using local authentication, let's analyze other user authentication options that could be applied using Lock-and-Key ACLs.

Lock-and-Key user authentication methods can be one of the following:

1. The password command, on the Virtual Terminal VTY line, which is a port authentication requiring the same password for all users establishing a Telnet connection to the NAS router. For obvious reasons, his authentication method is not recommended.
2. The username command from the local database of the NAS router, as explained in the last configuration. This authentication method has its pros and cons and is more recommended than the previous method, for its relatively better security aspects.
3. The Authorization, Authentication and Accounting (AAA) platform using an AAA external server like Terminal Access Control Access Control System (TACACS+) or Remote Authentication Dial In User servers (RADIUS) servers. This authentication method is sometimes referred to as "distant authentication", and is highly recommended, thus it will be explained in a following example.

All three-authentication methods mentioned above will be analyzed in details in chapter 5, entitled " Authentication and Dynamic ACLs". In the following section, we will demonstrate the authentication mechanism corresponding to each type of authentication method, while applied using Dynamic ACLs.

The authentication mechanisms used within Dynamic ACLs depend on the situations where Dynamic ACLs are applied. The situations, altogether with the

corresponding authentication mechanisms (E-Tutorials - Lock-and-Key Overview, 2012), are as follows:

I. When Dynamic ACLs are applied to allow/deny inner network temporary access to certain remote user(s) connecting through Telnet, the authentication of such users is either through the local firewall/router (NAS) or through another authentication server (AAA server). This application situation of Dynamic ACLs will be the focal point of this study.

II. When Dynamic ACLs are applied to allow/deny certain inner hosts to access a specific remote host/network, these inner hosts will be authenticated through an AAA server before allowing them to access the remote hosts/networks. This application situation is almost the opposite to the previous one.

III. When Dynamic ACLs are applied to allow/deny a remote user, by comparing his credentials to a list of credentials located on a local database attached locally on the NAS/Firewall router. Configuring the NAS for that purpose will be through the following command:
NAS(config)# **username** user1;
NAS(config)# **password** helloAccess;

Now that we introduced Dynamic ACLs using local authentication, as well as the different authentication methods that can be alternated and/or combined with the local authentication, it is worthwhile to introduce the authentication server authentication as a Dynamic ACL authentication method due to the wide use of this type of authentication, and due to the great security benefits it provides, as we will explain in details throughout the coming chapters. This type of authentication represents a distant authentication method, occurring on a device that is different than the NSA/firewall router (versus the local authentication method and the VTY authentication method with occur locally on the NAS router).

3.7 Dynamic ACLs and Authentication Servers

In this section we describe the authentication servers' concept in general and how they can interact, as a third authentication option, with Dynamic ACLs in order to obtain a secured remote user access. Thus we will start this section with a general overview about authentication severs followed a specific example of Dynamic ACLs using authentication servers as an example of a robust authentication method that can be used within Dynamic ACLs. This will bring us to the importance of using Dynamic ACLs (combined with the authentication servers) versus only the authentication servers' security, alone. And finally, we analyze the configuration codes of authentication servers as used with Dynamic ACLs to secure remote user access. The detailed analysis of such codes will be a cornerstone for further analysis of other authentication server concepts that will be studied in the following chapters.

3.7.1 Authentication Server overview

Authentication servers, which are sometimes called "authentication proxy"[23], were introduced in Cisco IOS software release 12.0.5.T to authenticate inbound and outbound users using a TACACS+ or a RADIUS AAA server[24]. The AAA servers can be used either by themselves or accompanied by dynamic ACLs, which is the case researched in our study. AAA servers depend on the AAA paradigm, which separates the Authentication, Authorization and Accounting functions, allowing different configuration for each function, which provides flexibility to allow a better control and more reliability of the network security process. These elements will be discussed in details in chapter 4 and 5.

In order to reach the inner resources of a network secured by a Dynamic ACL using an AAA server authentication, the remote users are originally blocked by the NAS' dynamic

---

[23] It is not recommended to use the « Authentication proxy» terminology in order to refer to AAA servers, as Authentication proxy mechanism is a little different than the AAA servers combined with Dynamic ACLs.

[24] Kerberos, as well as DIAMETER, are also used as alternative authentication servers, along with RADIUS and TACACS.

ACL, which only allows Telnet connections. Once that connection is established, a user has to authenticate through an authentication server in order to reach the organization's inner resources.

Authentication servers are based on a client/server architecture, where the client software resides on the network NAS/firewall router. Thus we will refer to the NAS router as "client". This router is in fact the access server situated at the boundary of the inner network, the one accepting access requests from remote users attempting to access the organization's resources located on the inner network. The Authentication Client Software, in some cases, can be distributed throughout the network. For the purpose of this study, we only consider the case where the client is located on NAS.

The server software runs on a computer typically situated within the inner network of the organization, which we will refer to as the authentication server (the AAA server). The server receives access requests sent from the NAS (the client), authenticates the remote users (referring to its local database) and returns some configuration changes back to the client or to another authentication server (that will return them eventually to the client).

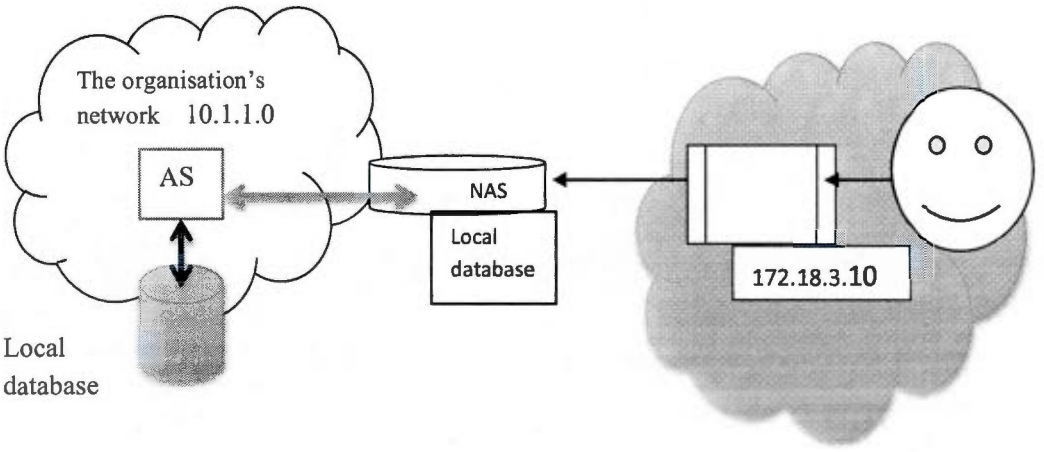

Figure 3.4 The Authentication server (AS) and the Dynamic ACLs User Authentication process.

According to figure 3-1, let's assume a certain user, using the IP address 172.18.3.10, needs to login remotely to the organization resources, represented by the network 10.1.1.0/24. The user establishes a Telnet connection, going through the organization's firewall, in order to authenticate through an authentication process done by the authentication server. The TACACS+ or RADIUS server, then, modifies the Lock-and-key ACL configuration, by sending additional ACLs entries to the NAS router (firewall) in order to allow the user's access once the authentication process is done successfully.

3.7.2 Dynamic ACLs need versus Authentication Servers

Now let's ask the question, why do we need Dynamic ACLs, in the first place, if we already have the option of using authentication servers?

The answer for this question lies in the capability of Dynamic ACL to create a temporary opening in the NAS router to allow access to the authenticated remote users, so that they can reach the organization's inner resources. Dynamic ACLs design allow choosing any of the authentication methods mentioned before (including authentication servers), according to their application situation, their reliability, efficiency, and many other factors that will be explained in details in chapter 5, entitled "Authentication and Dynamic ACLs".

Authentication servers by themselves help authenticate the user to allow resource access only to legitimate users. However that role doesn't specify a limited-time temporary access to the resources, that can dynamically terminate according to each resource specified access policies, as it is the case with Dynamic ACLs.

As a conclusion, in order to get a dynamic time-controlled access, Dynamic ACLs should be implemented to secure the remote access to an organization, whether using the authentication servers as user authentication methods, or not.

3.8 Dynamic ACLs Configuration (using Authentication Servers)

Now, let's think about the authentication server as an authentication method option to authenticate Dynamic ACLs' users, as shown by the following configuration:

> Dynamic ACLs configuration using proxy servers will follow the same configuration steps as the previous configuration, using Telnet authentication. These steps are as follows:
>
> Step1: Creating an extended ACL, permitting Telnet and specifying a placeholder entry for the dynamic entry that will be created through the autocommand.
>
> Step 2: Specifying the authentication method used, which is one of the three options already mentioned above.
>
> Step 3: Enabling that authentication method, on the router's VTY line so that the router is able to create dynamic entries through the autocommand, on its interface that has the Lock-and-key reference (E-Tutorials - Lock-and-Key Overview, 2012).

In order to configure the NAS router for Dynamic ACLs using TACACS+, the generic configuration code, according to Cisco (Cisco IOS Security Configuration Guide, Release 12.2-Configuring Authentication, 2006), is as follows:

```
nas(config)# aaa new-model                                    (1)
nas(config)# tacacs-server host IP_address                    (2)
nas(config)# tacacs-server key key                            (3)
nas(config)# aaa authentication login
authentication_name group tacacs+                             (4)
nas(config)# line vty 0 15                                    (5)
nas(config-line)# login authentication_name                   (6)
```

This configuration starts by enabling the AAA security services globally on the NAS router, in command (1).

Command (2) is to specify the TACACS+ server(s) IP address (es).

Command (3) is to specify the TACACS+ shared encryption key as a variable (called *key*).
Command (4) is to specify TACACS+ as the login authentication method, and the variable called "*authentication_name*" was substituted by the value called "group tacacs+", to indicate that the authentication method used is the TACACS+ authentication server, as we will see through the following detailed example. This command is also to enforce the use of the list of all TACACS+ servers located in the organization's inner network when authenticating the remote user login. This command line can be ended by the word "local", to indicate that in case the authentication through TACACS+ server(s) returns an error[25] during the authentication process, then the authentication will be attempted using the local database on the NAS server itself.
Command (5) is to enter a line VTY mode and, it is essential to include all VTY lines when authenticating Lock-and-Key access.
Command (6), which is the last line of the configuration, is to specify the authentication method used upon user login, and can be substituted by the following code:

```
nas(config-line)# login tacacs+
```

Now let's consider an example in order to show how Dynamic ACLs can be configured to use authentication server authentication.

[25] An error during the authentication process may occur because the TACACS+ server is down or cannot be reached, as well as any backup TACACS+ servers. An error is not the same as failed authentication which returns: access denied; in such case the user is denied the access and no attempts for further authentication are made by the NAS.

3.9 Example of Dynamic ACLs using authentication servers

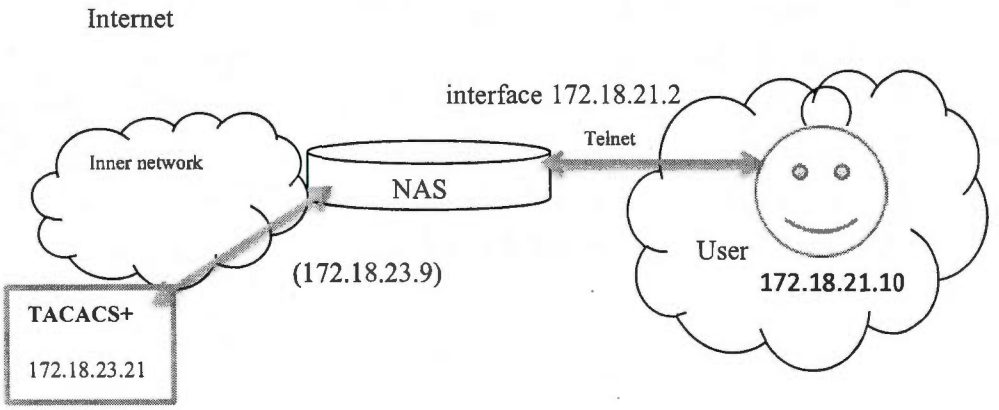

Figure 3.5 The TACACS+ server and the Dynamic ACL's User authentication process

The scenario through which the user will access the inner network resources will be almost the same as we saw with the last example using Dynamic ACLs with local authentication. According to figure 3-2, the remote user, say at IP address 172.18.21.10, will start his connection by establishing a Telnet session with the NAS interface 172.18.21.2. The Dynamic ACL will only allow such a VTY connection and the authentication method described within the VTY connection will be followed. The VTY connection will follow a default authentication method attempting to authenticate the user using the first TACACS+ server available 172.18.23.21. In case the authentication process is successfully performed by the TACACS+ server (which means no errors are returned from the server), and the user is successfully authenticated as well (which means his privileges allow him access to the inner resources and his credentials were correct), the user will be allowed access to reach the organizations inner resources allowed by the dynamic ACL.

This scenario might differ slightly in case the first TACACS+ server (172.18.23.21) is not available, in case the authentication using all TACACS+ servers returned errors, or in case the user could not authenticate, as his credentials are not valid. Such situations will be explained in details in chapter 4, entitled "User Authentication".

The configuration of Dynamic ACLs using authentication servers on the NAS router is represented by an example developed by Cisco (Cisco IOS Security Configuration Guide, Release 12.2-Configuring Lock-and-Key Security, 2006) within a specific implementation situation, where the network access is through ISDN. Hence the configuration includes a BRI-ISDN interface, in order to address such specific needs. The BRI-ISDN interface details are out of the scope of our study, thus we will not analyze such details amongst our in-depth analysis for the configuration code.

The NAS router configuration will be as follows:

```
aaa new model                                                    (1)

aaa authentication login default group tacacs+ enable            (2)

aaa accounting exec stop-only group tacacs+                      (3)
aaa accounting network stop-only group tacacs+                   (4)

enable password ciscotac                                         (5)
!
isdn switch-type basic-dms100                                    (6)
!
interface ethernet0                                              (7)
ip address 172.18.23.9 255.255.255.0                            (8)
!
interface BRI 0                                                  (9)
 ip address 172.18.21.1 255.255.255.0                           (10)
 encapsulation ppp                                              (11)
 dialer idle-timeout 3600                                       (12)
 dialer wait-for-carrier-time 100                               (13)
 dialer map ip 172.18.21.2 name Diana                           (14)
 dialer-group 1                                                 (15)
 isdn spid1 2036333715291                                       (16)
 isdn spid2 2036339371566                                       (17)
```

```
 ppp authentication chap                                          (18)
 ip access-group 102 in                                           (19)
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet         (20)
access-list 102 dynamic testlist timeout 15 permit ip any
any                                                               (21)
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2                   (22)
priority-list 1 interface BRI0 high                               (23)
tacacs-server host 172.18.23.21                                   (24)
tacacs-server host 172.18.23.14                                   (25)
tacacs-server key test1                                           (26)
tftp-server rom alias all                                         (27)
!
dialer-list 1 protocol ip permit                                  (28)
!
line con 0                                                        (29)
 password cisco                                                   (30)
line aux 0                                                        (31)
 line VTY 0 4                                                     (32)
 autocommand access-enable timeout 5                              (33)
 password cisco                                                   (34)
!
```

From the configuration code above, we can see that the Lock-and-key is configured directly on the NAS router, then it is applied to the BRI interface. The configuration commands are explained as follows:

- Command (1) corresponds to: enabling the AAA security services globally on the NAS router.

- Command (2), (3) and (4) correspond to: specifying the AAA functions to be used once a user logs in. In particular, once the user is logged in (via a Telnet connection), he will be authenticated through TACACS+ servers.
- Command (5) corresponds to: defining the enable password used on this router.
- Command (6) corresponds to: configuring the ISDN switch type, in order to match the service provider switch type. The ISDN switch is the same switch where the BRI 0 interface is located to connect to the ISDN line coming from the Internet.
- Commands (7) and (8) correspond to: defining the NAS router inner interface IP address (the Ethernet interface facing the organization's network).
- Commands (9) to (19) correspond to: defining the BRI NAS interface. Since the detailed examination of this portion of code is unnecessary in order to elaborate the configuration of Dynamic ACLs using AAA servers, we will explain it quickly without going through a lot of details. The explanation is as follows:
- Command (10) defines the interface IP address.
- Command (11) defines the interface encapsulation.
- Command (12) to (16) define the interface dialer, which is the remote host from which the remote users will dial in the BRI interface. Assigning the interface to a dial-in group helps controlling access into the interface. The mapping of the dial-in group into the BRI interface will transfer the functions of such group to the BRI interface. Thus, though the dynamic ACL is really to allow Telnet connection into the dial-in interface, it will be transferred into the BRI interface.
- Command (17) and (18) are to are to specify a service profile identifier and a local directory number to the B1 channel.
- Command (19) is to apply the Dynamic ACL 102 onto the BRI interface.
- Command (20) corresponds to defining a statement for access list 102, allowing only Telnet traffic to the dial-in interface.
- Command (21) corresponds to defining another statement for the same access list allowing only authenticated users (through TACACS+) to access any of the inner network resources for a maximum duration of 15-minute absolute timeout period.
- Command (22) corresponds to defining the routing path for network traffic.
- Command (23) corresponds to defining the BRI interface priority.

- Command (24), (25) and (26) correspond to defining both TACACS+ servers. Having two TACACS servers is helpful for redundancy purposes.
- Command (27) corresponds to defining the Transfer File Transfer Protocol (TFTP) server.
- Command (28) corresponds to associating the dialer group number with an access list number.
- Command (29) and (30) correspond to defining the password for the console connection into the NAS router.
- Command (31) corresponds to defining the auxiliary connection into the NAS router.
- Command (32) corresponds to defining the VTY connection.
- Command (33) corresponds to defining the autocommand to be triggered from within the VTY connection in case the user is successfully authenticated within a 5-minute idle timeout period.
- Command (34) corresponds to defining the VTY connection password.

The mechanism that takes place while a user tries to login is as follows:

1. The user at IP address 172.18.21.10 will attempt to access the BRI interface, the ACL defined on the interface (in command (19) will only allow him to Telnet to the dial-in interface, where he enters the VTY password specified in command (34) in order to establish a Telnet connection.

2. Once the connection is established, the user has only 5 minutes to enter his credentials for TACACS+ authentication before the Telnet session terminates.

3. The authentication method mentioned in command (2) specifies TACACS+ to be used, so TACACS+ servers (in command (24) and (25)) have to be tried one after another in case any returned an error. The command also mentions enable as a second authentication method, so the user has to be prompted for the enable

password in command (5) in case all TACACS+ servers are not available to perform the authentication. Once the user is authenticated successfully the autocommand in command (33) within the VTY connection will be triggered.

4. The autocommand will add a temporary dynamic ACL statement to ACL 102 mentioning the remote user IP address and the inner resources to be allowed to him, which are the same resources allowed by the original ACL statement (in command (21)). The statement will look like:
   "access-list 102 permit ip host 172.18.21.10 any"
   The statement will be placed after command (19) and before command (20).
   The user will be disconnected if he uses the organization's inner resources more than 15 minutes as mentioned in the absolute timeout of command (21).

Now that we explained Dynamic ACLs using both local authentication method as well as TACACS+ authentication as an AAA server authentication method, the reader can have a better understanding about dynamic ACLs in general, which was the main purpose of this chapter. The following chapter will consider user authentication methods in details, explaining how they take place and analyzing the main differences about their techniques.

In general, chapter 3 can be considered as a continuation of chapter 2, introducing one more type of IP ACLs. However, since this specific type is the main concern of the research, a whole chapter was needed in order to thoroughly study its mechanism, the way it is applied and the different options related to its application.

Thus the chapter provides an in-depth explanation of the concepts related to Dynamic ACLs through a deep theoretic analysis as well as through developing detailed configuration codes that will help draw the focus of the reader towards the smallest details of the implementation and the operation methodology of the concepts introduced, in order to better understand the mechanism behind each detail, and the potential security issues it might represent, in order to further emphasize these issues in the following chapters. Thus,

the chapter, in brief, introduces the Dynamic ACLs as a concept, stressing on its mechanism in general, as well as while using different authentication options.

In details, the chapter starts by providing a thorough explanation of Dynamic ACLs purpose, usage and mechanism. Then the chapter introduces the concept of Dynamic ACLs authentication, starting with the local authentication details, emphasized by a detailed configuration example; and ending by the authentication servers (external) authentication, followed by a comparison with authentication servers, in order to emphasize the importance of Dynamic ACLs creating a temporary access to the remote user; and emphasized by a detailed configuration example of Dynamic ACLs application along with authentication servers. Analyzing Dynamic ACLs along with its detailed operation and configuration along with the local and the external configuration method, in this chapter, helps understanding the detailed mechanisms of these concepts and thus allows for more analysis in the following chapters, in order to easily pinpoint the related security issues related to such concepts.

CHAPTER IV

USER AUTHENTICATION

4.1 The user authentication and the AAA paradigm

In a standalone, isolated system, identifying users is easily performed through recognizing the persons physically accessing the system, thus system data and resources are easily protected through the physical system protection procedures. However, in a time-sharing multi-user environment as well as a networked environment, other security procedures must be applied. User access rules must be set for each user, and thus, each user must be identified by secure means.

There are three approaches to implement on a given system once a user attempts to access it. These approaches are as follows:

1. User will be allowed access right away;
2. User will be allowed access only after providing his identity and will be trusted by the system;
3. User will be allowed access only after providing his identity and will be asked to prove it by the system;

The first approach is used in closed environments where all system machines are under control, of the persons physically situated in the system environment.

The second approach is used in more open environments for hosts (with known IP addresses) that are under organizational control (rlogin and rsh programs are used by these hosts' users).

The third approach is used in what's more likely the usual environment for today's systems, where being networked means having a functional/productive system. Thus the users on the network cannot be classified by their physical location or by their IP addresses (which are constantly changing), hence the need for the user authentication process in order to guarantee the security of computer systems.

Thus, Authentication is the process that insures that the person (or product) is who he claims to be, regardless of the access rights he's entitled to. According to the National Institute of Standards and Technology (NIST- Electronic Authentication Guideline, 2006) Authentication refers to three different classes/factors that can determine that a person is really who he claims to be:

1- Ownership factor: when the user owns something that would help him authenticate (ID card, wristband, security token, phone or cell phone)
2- Knowledge factor: when the user knows something that would help him authenticate (password, personal ID number);
3- Inherence factor: when the user is or does something that would help him authenticate (Biometric identifier, personal signature, bioelectric signals or a combination of these).

Two factor authentication: might require the user to use one of the ownership factors along with one of the knowledge factor. In other cases, where very highly secured systems are to be reached, a combination of biometric factors could be required along with one of the knowledge factor in order to grant access to the user.

Authentication is one of the three main processes to be done by an Authorization, Authentication and Accounting (AAA) paradigm, as it is obvious from its name. According to the IETF (IETF- RFC 2309- Generic AAA Architecture, 2000), the AAA paradigm provides three main network security features, needed to guarantee the safety of any remote access communication. The features are as follows:

1- Authentication, to specify which users are permitted access to the organization network, and to allow their access.
   User authentication takes place when a user first logs in to a network, and might be configured to take place only in cases where the user attempts to access certain types of services, network servers or extra user privileges.
   User authentication is a process upon which authorization (granting a privilege to the user), privacy (hiding information from unauthorized users) as well as non-

repudiation (the inability to deny an authorized action that has taken place based upon the authentication process) depend (Wikipedia, 2012).

2- Authorization, to specify which network resources are permitted for each user's access, once that user is authenticated. Usually the user Authorization process follows the user Authentication process. Since Authorization might not be configured, the user will be considered in this case, as having the same authentication as non-authorized users. However, in case the authorization is configured, while being preceded by a successful user authentication process, the user authorization will be applied according to the user corresponding profile. In some cases, the user profile might take place as a dynamic process when integrated with connection negotiations, like in PPP connections using TACACS server Authorization process, where both user authentication and user authorization processes are separately negotiated and integrated. In some other cases, user authentication and user authorization are both inseparably coupled using the user profile, like in the case of RADIUS server, as will be explained in Chapter 5.

3- Accounting, to monitor the network access and activities for all logged in users. The Accounting process usually follows the user Authorization process. It can be used for accounting/billing for the organization network services used or for network activity auditing and security. Certain fields can indicate the beginning, continuation or termination of the service to be billed.

The AAA paradigm enforces a distinction of the three AAA functions in order to guarantee the flexibility and reliability of the implementation. Since each function will take place independently and can be performed rigorously by different independent servers, each of which configured differently; or they can even be skipped sometimes, depending on the needs for the implementation situation. .

Such flexibility offered by the AAA paradigm, due to the separation of the AAA functions, would help us imagine the AAA paradigm as a framework or a cloud that encompasses the concept of independent security functions situated on one or more

authentication server (of the same type or different types), with which the NAS has to communicate in order to apply such security functions. Such a concept can be explained by figure 4-1, which represents the NAS relating to the AAA functions from a service perspective rather than from connection prospective.

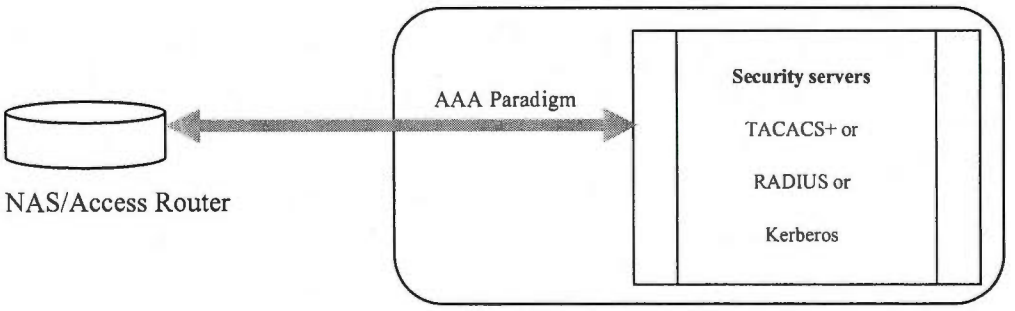

Figure 4.1 The relationship between the NAS and the security servers through the AAA paradigm

As shown by figure 4-1, The AAA paradigm is the means through which the organization can establish a secure communication between the Network Access Server (NAS/Firewall router) and the security servers. The AAA paradigm uses protocols like TACACS, Kerberos and RADIUS to administer its security functions during a connection between a client (like the NAS router) and any of the TACACS+, Kerberos, or RADIUS servers respectively.

As we introduced the three main functions of the AAA paradigm, it is important to know that the success of the Authorization and the Accounting processes totally depend on a successful Authentication process. Practically, once the user attempts to log into the network for a certain service, his identity is provided and approved by the authentication process, then his user privilege will be set according to that identity, specifying the types of services he's allowed to access through the Authorization process and then he can be billed for such services according to the amount and duration they were provided through the Accounting process.

Out of the three AAA functions, we will explain, in this chapter, the authentication concept, along with the different authentication methods (excluding local and line authentication[26]) that can be combined within Dynamic ACLs remote access process, in order to understand the different mechanisms they rely upon. Also we will demonstrate the means to combine them, and how to configure the NAS router in order to alternate between then as redundant security processes.

This chapter will help us better classify these authentication methods, compare them and suggest their combinations in different ways, within Dynamic ACLs, in order to ensure the redundancy of the authentication process.

The most important authentication methods that can be used to secure the Dynamic ACLs' remote access process is through the implementation of servers that abide to the AAA paradigm concepts. These servers are called " authentication servers". The authentication servers are to provide a proper user authentication, which would allow all network devices to trust that user's access. In the following we will describe the term "Authentication Server", then we will describe the different types of these servers.

4.2 Authentication server (AS)

The Authentication server (AS) is a central server, within an organization network, that is available to all the networked routers, switches and even servers for authenticating user. An AS is usually part of the AAA paradigm, thus providing the three main AAA functions for the networked devices using it. Also, the AS operates within the framework of client/server architecture, where the client is usually the NAS through which the remote user logs into the organization network, and the AS will authenticate that user in order to give him access to the network. This server is usually referred to as Authentication Server (AS), and it is the last device involved in the connection between an organization's inner network and the organization's remote users.

---

[26] Line authentication and local authentication will be explained in details in chapter 5, entitled "Authentication and Dynamic ACLs".

The following figure will help the reader better visualize the NAS, AS and client relationship; as follows:

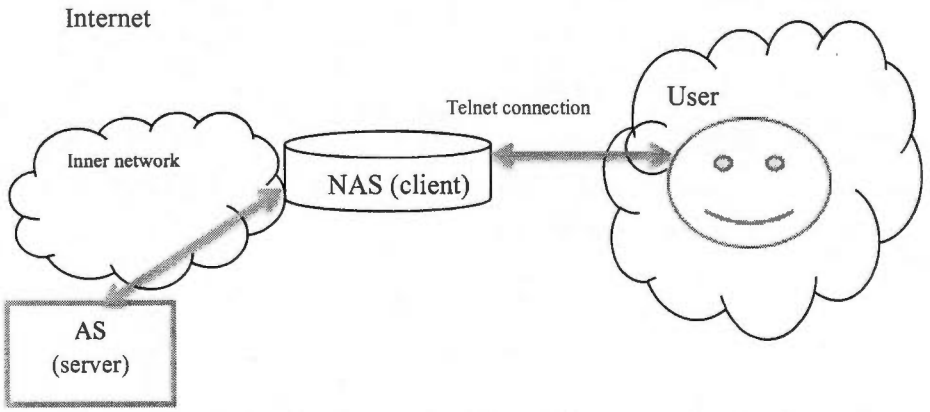

Figure 4.2 The Authentication server (AS) and the user authentication process

The AS can actually allow many remote users to connect to the same inner organization resource. However, since Dynamic ACLs mechanism doesn't allow securing a large number of connections, when combining AS within dynamic ACLs' authentication, securing connections will only be limitted to one remote user entry per each VTY line connecting to the inner network resources. Hence the number of connections can only be up to the number of the VTY lines. Dynamic ACLs nature is not scalable, that's the reason why it limits the scalability aspects of AAA servers[27].

A strong authentication, as approached and defined by many organizations, relies on a layered architecture of authentication points where many techniques are used to reach a certain authentication security level. Thus, an organization can use Dynamic ACLs as a first point of authentication, while combining them with other security techniques (like AAA servers) as further points of authentication. Thus AAA servers can be used as being both Dynamic ACLs authentication servers to authenticate the use access through the NAS router,

[27] The scalability subject will be further explained in chapter 5, entitled "Authentication and Dynamic ACLs".

on one hand, and/or as further authentication points after allowing user access through the Dynamic ACLs on the other hand. In this chapter we will exclusively focus on authenticating the user access through Dynamic ACLs using the AAA servers, using such servers as possible redundant authentication method in order to create an authentication error recovery process. In chapter 5, an example of AS as multiple-layer authentication using Dynamic ACLs will be introduced.

Now, we will introduce the different authentication servers used. However, in order to better understand the concept of authentication, there are some authentication approaches that sometimes are used within the mechanism of the authentication servers, and it is important to introduce these approaches beforehand in order to better understand the authentication servers' mechanism.

4.3 Approaches that are sometimes associated with the authentication process

The first and second approaches represent different mechanisms to receive the remote user credentials by the AS server; the third approach describes the layer-2 connection protocol that supports the fourth and fifth approaches; the forth and fifth approaches represent two authentication protocols, one of them applies the first approach introduced, and the other applies the second approach. The last approach represents a way to implement the AAA servers within the organization network.

These authentication approaches are as follows:

1- Challenge response.
2- One Time Password (OTP).
3- PPP
4- PAP.
5- CHAP.
6- Proxy servers.

4.3.1 Challenge/response

The authentication server (AS) can be configured to accept different types of user credentials, in order to authenticate the user. These types depend on the mechanism to be followed by the server to interpret and process the user's credentials. The two main types of user credentials are:

1- Regular passwords, which need less processing (than challenge/response) by the server. Some connection authentication protocols are based on this approach, like PPP-PAP, as we will introduce in the following section.
2- Challenge/response, which needs a certain mechanism to be interpreted and processed by the server. Some connection authentication protocols are based on this approach, like PPP-CHAP. In this section we will thoroughly explain that mechanism.

When the server is configured for challenge/response, the user supposed to authenticate using such a method, is usually pre-equipped with an external device on which he enters some numbers. The device performs some encryption for the numbers entered by the user. That device would be either a smart card or a security software, provided through the organization's network, in order to allow that encryption process.

The Challenge/response mechanism

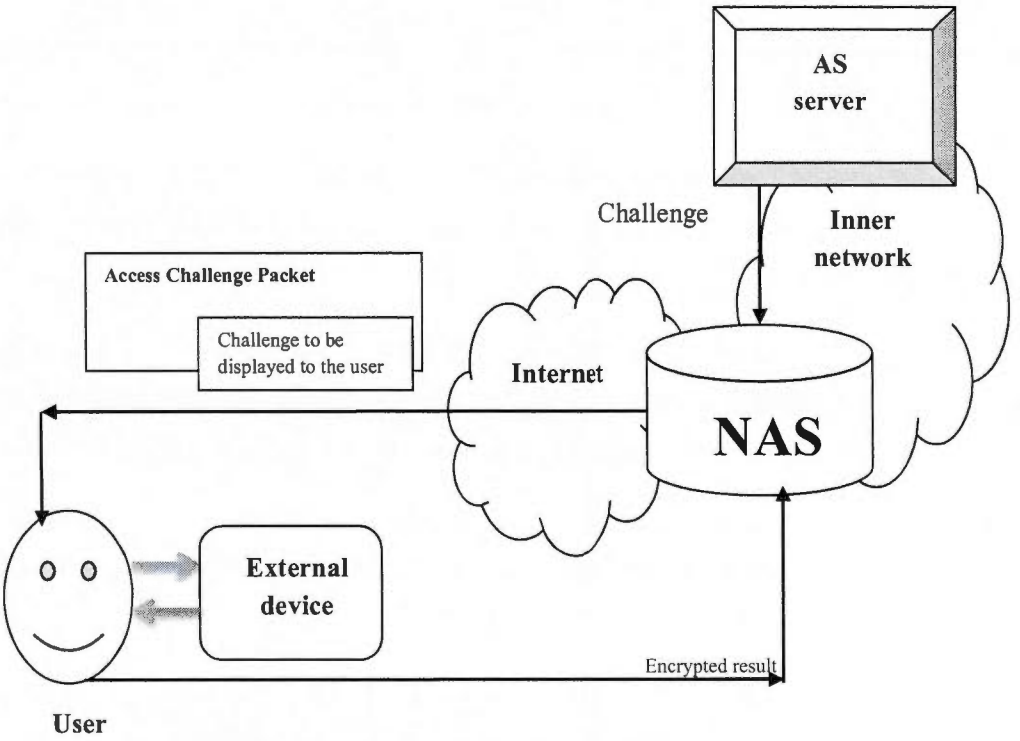

Figure 4.3 The Challenge Response flow

According to figure 4-3, when the user attempts to access the organization's inner network, he sends an Access Request to the NAS, and he receives an unpredictable number (challenge) that he's expected to encrypt (using an external device) and to send the encrypted result back to the authentication server (AS) through the NAS router, in the form of a new Access Request.

The challenge sent to the user is a non-repeating pseudorandom number, originally generated by an authentication server (AS), which knows the type of the authenticator device that the user uses to authenticate. Thus the challenge generated by the AS server changes every time the user attempts to authenticate; and since the credentials to be entered by the user (the response) are based on the value of the challenge encrypted through the user device, then these credentials will change also every time the user authenticates. In other

words, the user response continuously keeps adapting to the different challenges generated by the AS server.

The encrypted response provided by the user, which is the encrypted result calculated using the device, will be the first value that the server (AS) will use to authenticate that user.

The second value will be generated by the AS server, through performing the same encryption (as the one calculated by the user's device) for the challenge that was originally sent to the user.

If the encrypted response entered by the user (the first value) matches the encryption that the AS server expects (the second value), then an Access-Accept is sent to the user. Otherwise, if a wrong encryption is received, then the user gets an Access-Reject message for his access attempt through the NAS.

The steps for the challenge/response authentication process within the organization network boundaries are as follows:

1- The user attempts to access a resource belonging to an organization inner network, by sending an Access-Request packet to the NAS router.
2- The NAS sends an Access-Request packet to the AS server, including a request ID, a NAS Identifier and a port number as well as a user name and password (that were included in the user's Access-Request).
3- The AS server sends an Access Challenge Packet, including a State attribute and a Reply-Message, towards the user. The reply message is basically a text message to be shown to the user indicating his challenge number and prompting him to enter the encrypted response corresponding to that challenge. The state attribute helps the AS server correlate between the challenge number, sent to the user within the Access Challenge Packet, and the encrypted reply to be sent by the user to match the encryption of that challenge number.
4- The user uses his external device to get the encrypted response, and enters that response at the prompt (on the user client).

5- The NAS sends a new Access-Request to the AS server including a new request ID, a NAS Identifier and a port number as well as a user name and password. The latter is the encrypted response entered by the user. This new Access-Request also includes the same State attribute that was included in the Access-Challenge.
6- The AS server calculates the required value to which the user's response will be compared. Thus the AS server encrypts the challenge that was sent to the user, and keeps that value as a required value.
7- The AS server sends an Access-Accept or an Access-Reject based on whether the encrypted value entered by the user matches the required value. In some cases, according to each organization's need, the external server (RADIUS server) sends another Access-challenge, with a new number to encrypt.

4.3.2 One Time password (OTP)

A one Time Password (OTP) is another type password that is used only during one login session or one transaction, in order to avoid the security vulnerabilities of traditional static passwords. Though they require higher technology to generate and to implement, OTPs are reliable since they are only valid for a short time period.

OTPs are based on a time synchronization process between the client providing the password and the authentication server, and each OTP is generated as a new password based on a previous password using a mathematical algorithm involving a challenge (number) and/or a counter originated from the authentication server or the authentication process itself. Thus sequential OTPs are generated as a chain, and are used in a predefined order.

In order to inform the user about the next OTP to be used, new OTPs can be displayed on the user's security tokens that generate them, on the user's cell phone using a special software that generates them, on the user's cell phone while being sent through an SMS message as they are generated on the Authentication server side, or as a printed password on a paper carried by the user.

Now that we explained response/challenge as well as OPTs, let's introduce two authentication approaches/protocols that are sometimes used within authentication server mechanisms, including AAA servers. These layer-2 protocols are mainly used to authenticate the connection between the NAS as a client and the AAA server, in case when the AAA client/server connection is a layer 2-PPP.

These protocols are:

- ✓ Password Authentication Protocol PAP
- ✓Challenge Handshake Authentication protocol CHAP.

In this section, we will explain PPP connections before we highlight PAP and CHAP authentication methods in details.

4.3.3 Point-to-Point connections (PPP)

According to Tom's notes (IETF- RFC 1662, 1994), layer-2 Point to Point connections are link access procedures that are considered as one of the derivatives of the Cisco proprietary High-Level Data Link Control (HDLC) protocol, which is the main signaling standard used by WAN links.

Point to Point Protocol (PPP) has a layered architecture and is made of two sub protocols:

1. Link Control Protocol (LCP): to set up a PPP link.
2. Network Control protocol (NCP): To configure each of the OSI network layer protocols (for example, IP control protocol will be used to Control the OSI network layer IP protocol.

The LCP is responsible for PPP connection authentication, using PAP or CHAP, link compression, error detection, multilink support and load balancing, and PPP call back, which is requesting the other remote device to call back in order to establish the connection; thus it would increase the security of the link.

In order to establish a PPP connection, three steps have to take place in the following order:

- Link establishment: where LCP frames are used to configure the link.
- Authentication: as an optional phase where the link is authenticated for security, using PAP or CHAP.
- Network layer protocol: where the network layer protocols are configured. (IETF- RFC 1662, 1994)

Now let's explain PPP authentication protocols in details.

4.3.4 Password Authentication Protocol (PAP)

PAP is one of the two authentication protocols used with PPP connections, and it is based on the regular password approach. In order to establish a PAP authenticated session, the remote station's PAP keeps sending the username and password repeatedly until the NAS acknowledges it as correct, or otherwise terminates the session. According to E-How (E-How - What Are the Different Authentication Protocols, 2012), during a PPP-PAP connection, the username and password are sent in plain text, thus PAP authentication is not considered as a secure PPP connection.

To establish a PPP-PAP connection, once the NAS receives the PAP username and password, it includes them into an Access-Request packet that it sends to the AS server. The Access-Request packet also includes other attributes informing the AS server that a PPP service is expected.

These attributes are such as:

```
Service-Type=Framed-User
Framed-Protocol= PPP
```

Once the AS server receives this information, it looks up the user by the username in the server's database, and checks if the password sent by the NAS matches the password

expected. If match is found, the AS server sends an Access-Accept to the user through the NAS. Otherwise, an Access-Reject is sent.

4.3.5 Challenge Handshake Authentication protocol (CHAP)

CHAP is the second authentication protocol used with PPP connections. It is based on the challenge/response approach in order to authenticate a user or host (remote workstation) to an AS. CHAP authentication is based on a shared secret (like the client's user password)[28], which is always encrypted. CHAP also uses an incrementally changing identifier and a variable challenge value. In order to establish a CHAP authenticated session, a three way-handshake has to take place between the NAS and the remote station. It starts by a challenge sent from the NAS to the remote workstation's router. Then the router uses the user's password and the challenge received to calculate a checksum, using a one-way hash function to be sent back to the NAS. Then the NAS checks if the checksum is correct (according to the AS server authentication), and accordingly allows the connection, or drops it. There is always a limit for the numbers of trials received from the remote station's routers, which further limits the security risks. During a PPP-CHAP connection, the username, password and all data are sent between parties as an encrypted code, thus CHAP authentication is considered as a secure PPP connection.

CHAP authentication process can be detailed as follows:

To establish a PPP-CHAP connection, the NAS generates a random challenge to the user (CHAP Challenge). The user response includes a username, a CHAP ID and a CHAP challenge response. The CHAP ID and a CHAP challenge response represent the CHAP Password that the NAS would send, along with the username, within the Access-Request packet to the AS server. The Access-Request packet would also include other attributes informing the AS server that a CHAP service is expected. These attributes are such as:

---

[28] CHAP secret is referred here by the word "password" since it is mostly based on the user password, in an encrypted form.

```
Service-Type=Framed-User
Framed-Protocol= CHAP
```

Once the AS server receives this information, it looks up the user by the username and gets the corresponding password, and hashes it along with the CHAP ID and the CHAP challenge using MD5. Thus the user authentication dictates the presence of two main hashing processes, one is done by the user's router in order to conceal the entered user credentials (in a form of a checksum), and the other is done by the AS server in order to calculate a hash result based on the information already available in its database. This latter result is to be considered as a trusted reference, crucial for the authentication mechanism done by the AS for that specific user.

The AS server then, compares the hash result with the corresponding hash received from the user through the NAS. If a match is found, the AS server sends an Access-Accept to the user through the NAS. Otherwise, an Access-Reject is sent.

According to E-How (E-How - What Are the Different Authentication Protocols, 2012), CHAP requires that the user CHAP password to be available to the AS server in clear text format, so that the server would be able to hash the CHAP challenge and compare the result to the CHAP response.

In case where the AS server is unable to perform the requested user authentication, due to any reason, an Access-Reject is sent as authentication result.

Now that we explained the authentication protocols performed with PPP connections, let's introduce the concept of proxy servers. A proxy server, like the previous approaches introduced, is an approach that defines the security design and mechanism of a given network, and can be applied with all types of authentication servers. Therefore, it is important to thoroughly explain this approach before we get to explain the different types of authentication servers.

4.3.6 Proxy Server

Proxy servers are regular authentication servers, based on client/server architecture, where the NAS is the client. However, in the proxy server case, the NAS sends an access request to the forwarding server (an authentication server). The latter forwards the access request to a remote server (another authentication server), where the authentication takes place. An Access-Accept, Access-Reject, or an Access-challenge is generated by the remote server, and forwarded to the forwarding server, which will then forwards it back to the NAS. Proxy state attributes contained in the forwarded packets must not modify the forwarding server behavior, must not be modified by the forwarding server, and must not be omitted from the packets to be sent to the NAS (client).

Using proxy servers allows roaming, which is a feature permitting users of more than one device/server to connect to either device's network (realm) to get the same service at any point in time. Choosing the device/server that will receive the forwarded request, and thus will provide the service depends on either aspect of the following:

- The authentication realm, which may be a part of the network access identifier.
- The configuration of the forwarding server.

An authentication server can be both: a forwarding server (for some realms) as well as a remote server (providing authentication for other realms) at the same time. A forwarding server can forward access requests to one or more forwarding server, creating a chain of proxies, which ends with a remote server that provides authentication for any number of realms. In order to perform its functions securely, the forwarding server encrypts all messages received or transmitted; thus the server has to share a secret with the devices it forwards the authentication messages to and from. Thus, let's consider a network with a NAS router, a forwarding server and a remote server that authenticates the remote users logging through the router. The forwarding server will share a secret with the NAS router as well as different secret with the remote server, in order to avoid forwarding the user authentication messages to rogue devices. Before forwarding the user authentication request and response between the NAS and the remote server, the forwarding server has to authenticate the messages with the different shared secret corresponding to each device.

In more details, proxy servers function according to the following steps:

1- The NAS sends an encrypted Access-Request to the forwarding server.

2- The forwarding server, already having a secret shared with the NAS, decrypts the user password using that shared secret. Any other attributes must not be modified by the forwarding server, from point of view contents, or place-order in the packet. The forwarding server might be configured not to send some attributes to the remote server, however all state attributes must be sent back to the NAS. The forwarding server might choose to add one (and no more than one) more proxy-state attribute to the packet.

3- The forwarding server already having a secret shared with the remote server re-encrypts the user password using that shared secret, and forwards the Access-Request to the remote server (or to other proxy forwarding servers).

4- Once the remote server becomes the final destination, it authenticates the user using the user password or the CHAP password and returns the authentication result: Access-Accept, Access-Reject, or an Access-challenge, included in a Response-Packet to the forwarding server.

   All Proxy-State attribute that were originally included in the Access-Request that reached the remote server must be copied in the same order into the Access-Response to be sent by the remote server to the forwarding server.

5- Once the Response-Packet reaches the forwarding server, the latter authenticates the remote server first in order to make sure it is not a rogue server, by verifying the Response Authenticator attribute, using the secret shared with the Remote server. In case that the authentication fails, the Response-Packet will be discarded. Otherwise, the forwarding server will remove the last Proxy-State attribute, sign[29] the Response Authenticator, encrypt the transmitted packets using the secret it shares with the NAS (so that the NAS would not identify the forwarding server as a rogue server

[29] When a device signs a packet, it is to prove the device legitimate identification for the other devices that will handle that packet latter, so that these devices won't consider the original device as illegitimate.

either), restore the Identifier which relates the Access-Response to the original Access-Request sent by the NAS, and finally send the access-Response to the NAS.

Now that we explained the different security approaches to be used with authentication servers, let's introduce in this section, some of the AAA authentication servers/protocols in details.

4.4 AAA authentication servers/protocols

The protocols introduced are:

1. RADIUS.
2. DIAMETER.
3. TACACS+.
4. Kerberos.

4.4.1 Remote Authentication Dial In User Server - RADIUS

According to Wikipedia's definition ((Wikipedia - RADIUS, 2012), RADIUS[30] is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.

RADIUS, as an authentication protocol, is specified by RFC 2865 and RFC 2866. Many commercial and open source RADIUS implementations are available on the market. Though it was originally developed to authenticate dial-up connections, RADIUS was adapted to authenticate most types of connections that can be established to the organizations' network resources, including layer 3 connections. Also because of the broad support and the ubiquitous nature of the RADIUS protocol, it is often used by ISPs and enterprises to manage access to the Internet or internal networks, wireless networks, and

[30] RADIUS was developed by Livingston Enterprises, Inc., in 1991 as an access server authentication and accounting protocol and later brought into the Internet Engineering Task Force (IETF) standards.

integrated e-mail services. These networks may incorporate modems, DSL, access points, VPNs, network ports, web servers, etc.[31]

RADIUS is implemented via UDP Port 1812, between a Network Access Server (NAS or client) that needs to authenticate its links and a shared Authentication server (the RADIUS Server). RADIUS client components might reside on any of the access points (gateways) of an organization's inner network, such as a Network Access Server (NAS), a Virtual Private Network (VPN) server, a network switch with port based authentication, or a Remote Access Server (AS).

RADIUS user authentication process and user authorization process are inseparably coupled, as well as they take place while using the policies within the user profile. Thus once the user logs into the network, he gets authenticated and his profile dictates which types of network resources and services he is permitted to access.

RADIUS, as an authentication protocol, can look up users in text files, Lightweight Directory Access Protocol (LDAP) servers, and various databases. Although RADIUS can be used for authorization and accounting, as mentioned above in the study, we will only describe RADIUS as an authentication protocol.

Implementing RADIUS, as an authentication technique, usually takes place while managing a single database of users containing the users authentications (username and password and/or challenge/response) as well as the specification of the type of service entitled to them (example: SLIP, Telnet, rlogin,..).

RADIUS is a transaction-based protocol, which relies on UDP to establish the connection between the client (NAS) and the server. RADIUS requires the following:

1- Allowing the Access-Request to be transmitted to a secondary server in case the Primary authentication server fails.

---

[31] Defined in RFC 3588, DIAMETER is the upgrade authentication protocol for RADIUS. DIAMETER will be introduced later in this study.

2- RADIUS protocol doesn't require responsive data loss detection, so it doesn't require the usage of TCP with its related acknowledgement overhead.
3- The user, to be authenticated using RADIUS, doesn't need to wait a longer time while the reliable TCP protocol reflects the authentication result, while he can use a faster alternate server by repeating the authentication attempt. Thus, UDP makes sense to be used as a faster transport layer protocol.
4- The stateless nature of RADIUS makes UDP a better choice, as no need to detect lost connections. The RADIUS servers UDP connections to Clients are opened once and stay open for simplicity.
5- UDP handles the transport of many processes to answer a client Access-Request better when processed by a multi-threaded server like RADIUS.

RADIUS depends on the client/server architecture where the NAS is the client that needs the RADIUS server to perform the user authentication. Thus, the NAS hands the RADIUS server the user login information as a request for authentication, and waits for the authentication results coming back from the server as a response, along with some configuration details. Upon receiving such response, the NAS becomes ready to take some actions, and to deliver a certain service to the user accordingly.

According to RFC 2865 draft Standard (Rigney, Willens, Rubens, & Simpson, 2000), RADIUS supports a number of authentication methods, like PPP PAP, PPP CHAP, UNIX login and other authentication mechanisms. The NAS should only ask for access authorization depending on the type of access services it provides. NAS provides PPP and Telnet services for dial-in users. If an authentication for an access service not provided by the NAS is received by the NAS, the NAS should reject that authentication since that NAS doesn't provide that access service.

RADIUS may act as a proxy client for other servers, either RADIUS or other authentication servers, as well. However, for the sake of simplicity, during this study, we focus on the model where there is only one RADIUS servicing one NAS.

RADIUS mechanism

RADIUS mechanism can be described according to its draft RFC (Rigney, Willens, Rubens, & Simpson, 2000), through the following steps:

- Users, desiring a connection session(s) into the organization's network, provide their login information to the NAS (client) through a customizable login prompt, allowing them to enter their credentials[32], or through a link framing protocol like Point-to-Point (PPP) protocol that has its own authentication packets to carry such login information.
- The login information (user credentials) communicated between the NAS and the RADIUS are always secured. This information can take of two possible forms: in the form of passwords that are communicated between the NAS and the RADIUS server, while being hashed (usually through using the Message Digest Algorithm MD5); or in the form of shared secrets that are never sent over the network. This eliminates the chance of malicious snooping and security attacks over the network connections.
- Then the NAS sends that login information (received from the remote user) to the RADIUS Server in order to authenticate the user, the user credentials will be sent in the form of an "access request". The login information usually includes the user credentials, and the client ID and the port through which the user is accessing the network. If the NAS receives no response (from the RADIUS server) for the access request authentication within a certain time, the authentication request has to be resent to the RADIUS sever.
- Upon receipt of the authentication request by the RADIUS server, the latter validates the client (the NAS, in this case). Thus the RADIUS server has to have a valid shared secret for that NAS client, or the request will be discarded.
- The RADIUS server, then, verifies information sent by the client, so the user's credentials have to be identical to the user's information located within the RADIUS' database, as one condition to authenticate the user. Also the NAS and

[32] Credentials can be username/password and/or challenge/response.

NAS access port through which that user attempts to connect to the organization's network have to be identical to the NAS information within the network topology located within the RADIUS' database, as another condition to authenticate the user.

- If any of the conditions mentioned above is not met, the access request is rejected and the message is sent from the RADIUS, through the NAS, to the user. If all conditions are met, and the RADIUS is configured to challenge the user using a challenge/response mechanism: the RADIUS sends a challenge response to the user through the NAS. After receiving the challenge response from the user, the NAS sends it back to the RADIUS server in a new request ID, replacing the user password by the challenge response that came from the user, encrypted; as described in the challenge/response section above.
- The RADIUS server response to this request is either: access-accept, access-reject, or another access-challenge. In case of access-accept, the RADIUS response to the NAS will include the type of service to be given to the user (EX: SLIP, PPP, and Login User), along with other values that would allow the NAS to deliver the desired service (e.g., IP address, subnet mask, MTU, desired compression, desired packet Filter identifier, network protocol, host).

4.4.2 DIAMETER

With the emergence of new telecommunication technologies, including; wireless networks and IP mobility, the requirements of the AAA framework as well as user access control mechanisms have increased in complexity and diversity. However, RADIUS as an AAA protocol was not sufficient to fulfill such emerging needs, thus DIAMETER was developed by an IETF team, mainly for the US military, in order to cope with the access control features, with the flexibility to be extended in order to support new functionalities.

Defined in RFC 3588, DIAMETER was developed by Pat Calhoun, Glen Zorn and Ping Pan in 1998, for providing newly needed requirements as an AAA framework and as an upgrade authentication protocol for RADIUS. Though it is not compatible with RADIUS, DIAMETER was introduced to overcome RADIUS issues related to reliability, scalability,

security and flexibility, as well as handling remote access, IP mobility and policy control. In this study we will elaborate DIAMETER only from the user authentication point of view.

According to Wikipedia (Wikipedia - Authentication Protocols, 2012): "since it relies on TCP and SCTP (Port # 3868) as more reliable Transport layer protocols and on IPsec and Transport Layer Security (TLS) versus unreliable UDP Transport layer protocol and only TLS in the case of RADIUS, DIAMETER is considered as a more robust type of RADIUS server. Especially that it is further enhanced with the introduction of the 3rd Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS) as well as with its applications support to different types of interfaces and its extensibility, which allows the support for Proxies, Brokers, Strong Security, Mobile IP, Network Access Servers (NASREQ), Accounting and Resource Management."

Other advantages of DIAMETER over RADIUS, according to Wikipedia (Wikipedia - Authentication Protocols, 2012), include the following:

- Larger space for Attribute Value Pairs (AVP)[33] and application Identifiers.
- Both stateful and stateless models can be used.
- Dynamic discovery of peers.
- Capability negotiation.
- Support application layer acknowledgements and defines failover methods (RFC 3539).
- Error notification.
- Better roaming support.
- Basic support for user session and accounting.

According to Wikipedia (Wikipedia - Authentication Protocols, 2012), "DIAMETER applications" is a terminology referring to the protocols that are based on

[33] AVP defines DIAMETER protocols (applications).

DIAMETER, and it doesn't refer to software applications. A DIAMETER application should be defined by an Identifier and can add new command codes and a new mandatory Attribute Value Pairs (AVP)(Wikipedia - Authentication Protocols, 2012).

4.4.3 Terminal Access Control Access-Control System Plus (TACACS+)

TACACS+[34] is another prominent security client/server protocol that controls access into networks, and that follows the AAA paradigm. TACACS+ is the latest generation of TACACS and was developed by Cisco after RADIUS in order to cope with the ever-growing security market, as it is more scalable and more adaptable to new security technologies.

TACACS+, as an AAA server protocol, separates the AAA functions so that they can be implemented on three different servers, where each function provides a unique service and when all functions are combined they provide a powerful network protection. In fact, TACACS+ separates the user authentication and the user authorization processes while other AAA protocols/servers, like RADIUS, combines them into one user profile. Thus, the implementation of TACACS+ doesn't dictate using all three functions; such separation allows choosing the specific functions needed according to each network design targets, though all three functions can be provided by TACACS+ if needed.

And According to the International Engineering Task Force (IETF) (IETF-TACACS+ Internet Draft, 1997), when it comes to the user authentication, which was always linked to the user authorization in previous TACACS generation, it is now a separate process, which allows dynamic user authorization independently from the user profile. So

[34] According to Wikipedia (Wikipedia, 2012), TACACS was originally developed by BBN in 1984 for MILNET, the US. Department of Defense with the purpose of automating login onto a network device for a user once he's already authenticated to login onto another device located onto the same network. Over the 1980s, Cisco added some extensions to TACACS and it was called XTACACS. Then in 1996, TACACS+ is the Cisco proprietary enhancement of the original TACACS, however it had no official RFC documentation. TACACS+ is allowed on port 43 on TCP/IP connections.

instead of a one standard user profile, TACACS+ can involve other negotiations, such as PPP, for more flexible user profile that could be per-access user profile. Separating the AAA functions has improved the Accounting portion as well, as it took it to the next level of security auditing and billing services. TACACS+ allows the client to send very fine-grained access requests and allows the server to respond to every part of that request.

The authentication types supported by TACACS+ are the following:

- ASCII
- PAP
- CHAP
- ARAP
- MSCHAP

TACACS also supports Forwarding Proxies (already explained above).

In this section, we focus on the authentication function of TACACS+, as the Authorization and the Accounting function details are out of the scope of this study.

Another interesting feature of TACACS+ according to The IETF (IETF-TACACS+ Internet Draft, 1997), is that it encrypts all traffic between the client, which is the Network Access Server (NAS), and the TACACS+ server (the AS server performing the TACACS+ authentication). TACACS+ allows flexibility regarding site customization, future development features to be added as well as multiprotocol support like IP and AppleTalk (as Network layer protocols).

According to Cisco (Cisco-TACACS+ and RADIUS Comparison, 2008), TACACS+ uses the transport layer protocol: Transmission Control Protocol (TCP), as opposed to TACACS and XTATACS that use UDP. Using the reliable, connection-oriented transport TCP protocol has many advantages over using the best effort-delivery UDP protocol:

1- Every time a request is received, TCP has to provide an acknowledgement from the receiving device to assure that the packet has been sent to its destination, no matter how busy the TACACS+ server might be.

2- TCP connections give indications about crashed, slowing down, shut down, non-existent or up and running servers, while UDP doesn't differentiate between the different servers status.
3- The management of many server connections is more easily performed, and simultaneously maintained, while allowing the detection of crashing servers, allowing the possibility of sending messages only to functional servers.
4- TCP is scalable, reliable and works well within congested networks.

TACACS+ and Authentication

Authentication configuration is not mandatory for TACACS+ functionality, as it is considered an optional security measure that depends on a given network's design goals and organization's security policies. Some networks/sites might incorporate TACACS+ with no authentication configuration; some other sites might only require user authentication in case the user attempts to access certain types of services, to reach certain network servers or to gain extra user privileges. In any of these cases, TACACS+ would perform its AAA functions properly.

User authentication can take many forms. The most popular authentication form is a username and a fixed password, though this authentication form has its own drawbacks. TACACS+ permits such an authentication form; other more secure authentication forms (like the one-time password or challenge/response queries) as well as potential future authentication forms are supported as well. Also, TACACS+ allows more types of Authentication requests and more types of Authentication responses, than its predecessors TACACS and XTACACS.

An authentication session is a single authentication sequence of encrypted packets, an interaction between the client and the TACACS+ server, identified by a session ID to be known between the client and the TACACS+ server and to be included within the encrypted connection. Many sessions may be multiplexed over the same TCP connection if both the client and TACACS+ server support such multiplexing. Otherwise, a new TCP connection should be opened at the beginning of each session and closed at the end of the session. In

case of Authentication sessions, this process involves the exchange of an arbitrary number of packets, compared to only a couple of packets (a request and a reply) in the case of Authorization and Accounting sessions. In this study, we focus on highlighting the Authentication sessions, including the different activities that take place within. The Authorization and Accounting sessions are out of the scope of this study.

The session ID doesn't change through the whole duration of a given session and it has to be a cryptographically strong random number, otherwise it might compromise the packets' encryption security. Only one packet encryption mechanism should be used during a single session. This mechanism could rely on a secret value, which is a shared secret between the client and the TACACS+ server; or it could rely on a different key per client or TACACS+ server they communicate with. The latter option should be more secure if available; however such configuration choices are optional decisions to be taken by each organization network designers (IETF-TACACS+ Internet Draft, 1997).

The packet's body is encrypted by XOR-ing it with a pseudo-random pad, which is generated by concatenating a series of MD5 hashes. The first hash is generated based on the concatenation of the secret key (shared between the client and the TACACAS+ server), the session ID (in network byte order), the version number of TACACS+ (major and minor version numbers combined), and the sequence number (of the packet in that session). The latter three items are all included in the packet header. Each subsequent hash concatenation would include the same four values, just mentioned, all concatenated with the result of the previous hash (IETF-TACACS+ Internet Draft, 1997).

The authentication process starts by the following steps, according to the IETF Internet Draft (IETF-TACACS+ Internet Draft, 1997):

1- The client sends a START message to the TACACS+ server, including the type of the authentication to take place, a username and some authentication data. The start message is the first packet to be sent when starting a communication between a client and a TACACS+ server or in case of a restart; so it has the sequence number=1, since it is the first packet within a given session.

2- The TACACS+ server should send back a REPLY message indicating if the authorization process is finished (by a terminating REPLY of either PASS or FAIL), or it should continue, by asking (prompting the remote user) for more authentication information within the reply (GETDATA, GETUSER, GETPASS to get data, a username or a password respectively)[35].

3- In the latter case, the client will send the required information (coming originally from the user) in a CONTINUE message, to the TACACS+ server.

4- For every START or CONTINUE message (only sent by the client), the TACACS+ server has to send a REPLY message (only sent by the TACACS+ server); unless the client message includes an ABORT field in the CONTINUE message where the session should be immediately aborted, with no message sent by the TACACS+ server. The different messages exchanged for different remote access requests will be elaborated in the following section.

The types of message sent by the NAS (client) could be one of the following options, according to the IETF Internet Draft (IETF-TACACS+ Internet Draft, 1997):

1- An ENABLE request, to change the current privilege level of a principal[36]. In this case, many messages might get exchanged between the client and the TACACS+ server; so that the TACACS+ server would get the information it needs in order to change the privilege. This exchange is very similar to an INBOUND ASCII LOGIN.
2- An INBOUND ASCII LOGIN, where the START packet sent by the NAS may contain the username (not required), and might be followed by zero or more pairs of REPLY (sent by TACACS+ server) and CONTINUE (sent by client) in case the TACACS+ server needs more information, and then will be followed by a terminating REPLY of either PASS or FAIL (sent by TACACS+ server).

---

[35] Sometimes the TACACS+ server might send a RESTART ERROR message, or a FOLLOW message as well.

[36] A principal is a term defining an authenticated party (user or device).

3- An INBOUND PAP LOGIN, where the START packet coming from NAS contains the username and the ASCII password, and then the START packet will be followed by a single terminating REPLY of either PASS or FAIL (sent by TACACS+ server).

4- An INBOUND CHAP LOGIN, where the START packet sent by the NAS may contain the username (secret) and the concatenation of the PPP ID, the challenge and the response; and then the START packet will be followed by a single terminating REPLY of either PASS or FAIL (sent by TACACS+ server). To perform the authentication, the TACACS+ server runs MD5 on The PPP ID, the user secret and the challenge; and then compares the result to the response, to check if they match.

5- An INBOUND MS-CHAP LOGIN, where the START packet sent by the NAS may contain the username (secret) and the concatenation of the PPP ID, the MS-CHAP challenge and the MS-CHAP response; and then the START packet will be followed by a single terminating REPLY of either PASS or FAIL (sent by TACACS+ server). To perform the authentication, the TACACS+ server runs a combination of the cryptographic hash function Message Digest Algorithm (MD4) and the symmetric encryption algorithm Data Encryption Standard (DES) on the user secret and the challenge; and then compares the result to the response, to check if they match.[37]

6- An INBOUND ARAP LOGIN, where the START packet sent by the NAS may contain the username (secret) and the concatenation of the NAS challenge to the remote peer[38], the remote peer challenge to the NAS, and the remote peer response to the NAS challenge; and then the START packet will be followed by a single terminating REPLY of either PASS or FAIL (sent by TACACS+ server). In case of a PASS REPLY packet, it will include an encrypted Peer Challenge. To perform the authentication, the TACACS+ server runs a DES encryption on both challenges using the user secret as an encryption key; and then compares the result to the peer's response, to check if they match.

---

[37] An OUTBOUND MS-CHAP LOGIN is an example, among others, of situation the NAS needs to provide MS-CHAP authentication credentials to the remote PPP peer. Such authentications are out of the scope of this study, so will not be explained in this study. For a full list of such authentications, interested readers may refer to TACACS+ draft (IETF-TACACS+ Internet Draft, 1997).

[38] "Remote peer" is another name referring to many local and/or remote authentication servers working alternatively for redundancy.

7- An ASCII CHANGE PASSWORD REQUEST, where there would be multiple message exchanged between the TACACS+ server and NAS as the TACACS+ server would collect the information needed in order to change the user password.
8- A PPP CHANGE PASSWORD REQUEST, are not valid since PPP protocol doesn't support password changing.
9- An ARAP CHANGE PASSWORD REQUEST, where the START packet sent by the NAS may contain the username (secret) and both the old and new password encrypted; and then the START packet will be followed by a single terminating REPLY of either PASS or FAIL (sent by TACACS+ server).
10- An ABORT field in the CONTINUE message, and it might be followed by the reason why; the session drops without a REPLY message needed from the TACACS+ server.

The status of REPLY message sent by the TACACS+ server (server) could be one of the following options:

1. PASS, is sent when the user authentication takes place successfully;
2. FAIL, is sent when the user authentication doesn't takes place successfully, or doesn't take place at all;
3. FOLLOW, is sent when the authentication session should be terminated, regardless of the authentication action or type, and usually the sent packet includes a message to be displayed to the user. When sending a FOLLOW REPLY message, TACACS+ TACACS+ server request, at the discretion of the client, to switch the authentication process to be performed by an alternative TACACS+ server, described in ASCII text, as well as, optionally, by an alternative suggested authentication method (other than TACACS+). If no alternate host is chosen as a TACACS+ server, the authentication returned status is as a FAIL.
4. ERROR, is sent when the authentication session should be terminated, regardless of the authentication action or type, and usually the sent packet includes a message to be displayed on the administrative console or log. When sending an ERROR

REPLY message, TACACS+ TACACS+ server indicates that the alternative host couldn't be reached.

5. RESTART, is sent when the authentication session should be terminated, regardless of the authentication action or type, and usually the sent packet includes a message to be displayed to the user. When a RESTART REPLY message is sent, TACACS+ server indicates that the Authentication value (indicating the authentication type) included in the START packet sent by the NAS is not acceptable to be negotiated with the alternate hosts, however other START packets might be acceptable, that's why a RESTART action is requested so that the NAS would be able to negotiate the choice of such authentication types with the alternate peers (IETF-TACACS+ Internet Draft, 1997).

4.4.4 KERBEROS

Designed by Steve Miller and Clifford Neuman, Kerberos is a network security protocol, based on the client server architecture. The Massachusetts Institute of Technology (MIT) developed Kerberos to secure network services of project Athena, in the 1070s.

Version 5 is the last version of Kerberos made by John Kohl and Clifford Neman and documented in an RFC 4120 in 2005. In 2007, the Kerberos Consortium was formed, with many vendors such as Oracle, Apple Inc, Google, Microsoft, Centrify Corporation, and TeamF1 Inc., and many academic institutes such as KTH-Royal Institute of Technology, Stanford University and MIT sponsored it.

The latest updates done by the IETF Kerberos working group are documented into RFC 3961, RFC 3962, RFC 4120, RFC 1510, and RFC 4121. Kerberos provides a free, secure, reliable, scalable and user transparent authentication solution and is considered as a standard for windows 2000 networks.

Kerberos and authentication

Kerberos is a client server architecture whose idea depends on authenticating both the user as well as the server (mutual Authentication)[39], in order to prevent rogue servers from identifying themselves as legitimate ones, and then taking control of the system. Kerberos authentication depends on a Kerberos-proprietary certificate model, where an Authentication server (AS) performs the authentication process, while issuing permission tickets to its clients in order for them to grant user access to some other inner servers/services.

According to Steiner (Steiner, Neuman, & Schiller, 1988), the AS connects to a database that contains client information and private keys[40], on the organization's inner network, as well as being connected to the application server, which would be accessed by the user, on one hand. On the other hand, the AS connects to a NAS as a client through which the remote user connects to the network.

Like RADIUS and TACACS+, Kerberos authentication process for a user is through one whole connection session, however this access permission is not for all application servers situated on the network, it is only to the service to which the permission ticket was issued. In case other services need to be accessed; the client has to go through another authentication process to get a permission ticket that corresponds to that specific service. Thus when a remote user attempts to access an inner server, Kerberos has to provide a ticket to that server corresponding to that user as well as a proof that this ticket was not stolen.

During one session between two authenticated parties (principals), Kerberos can provide the option of generating temporary private keys (called session keys) to ensure a communication encryption throughout that connection session.

---

[39] Both users (clients) and servers to be authenticated by Kerberos are known as "principals".

[40] Kerberos uses private Key encryption, where each principal is assigned a large number, only known to that principal and to Kerberos. In case when the principal is a user, the private key is a user password.

There are three levels of Kerberos authentication, according to Steiner (Steiner, Neuman, & Schiller, 1988), as follows:

1. Authentication that happens with the connection initiation, assuming that all consecutive messages will originate from the same authenticated user. This level of authentication is used by the Kerberos File server as well as other application servers;
2. Authentication that happens with the connection initiation, combined with an authentication for every communication message originated by the same authenticated user. This level of authentication is used by Kerberos application servers, which don't care about the disclosure of the message contents, however they want to make sure all messages are originated from the same authenticated principal. These authenticated messages are then called: Safe messages;
3. Authentication that happens with the connection initiation, combined with an authentication for every communication message originated by the same authenticated user, as well as message encryption. This level of authentication is used by the Kerberos application servers that do care about the security of the message contents, besides making sure all messages are originated from the same authenticated principal. Kerberos uses this authentication level when exchanging passwords and private keys. These authenticated encrypted messages are then called: Private messages;

The choice of the different levels of authentication is to be determined by the network designer according to the connections security needs of every application server located on the network.

According to Steiner (Steiner, Neuman, & Schiller, 1988), Kerberos authentication process is based on Needham and Schroeder key distribution model and its target is to generate a ticket that authenticates the remote user to the end server. The process takes place through the following phases:

- Kerberos creates user credentials to be used with access requests to different services.

Credentials are tickets including the remote user password and authenticators. Tickets are used to pass the user identity between the AS and the End server, as well as to ensure the identity of the user. A ticket is good for a single service and a single client, however it can be used many times until it expires. The authenticator is for proving the user's identity. It contains information that, when compared with the ticket's information, can prove that the client sending the ticket is the same as the one to which the ticket was issued. The authenticator can be used only once. Both the tickets and the authenticators are based on private key encryption; however they are encrypted using different keys.

- The user requests authentication for a certain service.
  The Ticket Granting Ticket (TGT) lifetime is the duration through which the Kerberos client will consider a certain user as authenticated to access the organization's network to use a certain service. In order to obtain an access ticket for a certain end server, the AS has to communicate with the Ticket Granting Server (TGS) that generates such tickets. That communication is by sending a Ticket Granting Ticket (TGT) to the TGS.
  Kerberos authentication processes can be illustrated by figure 4-4.

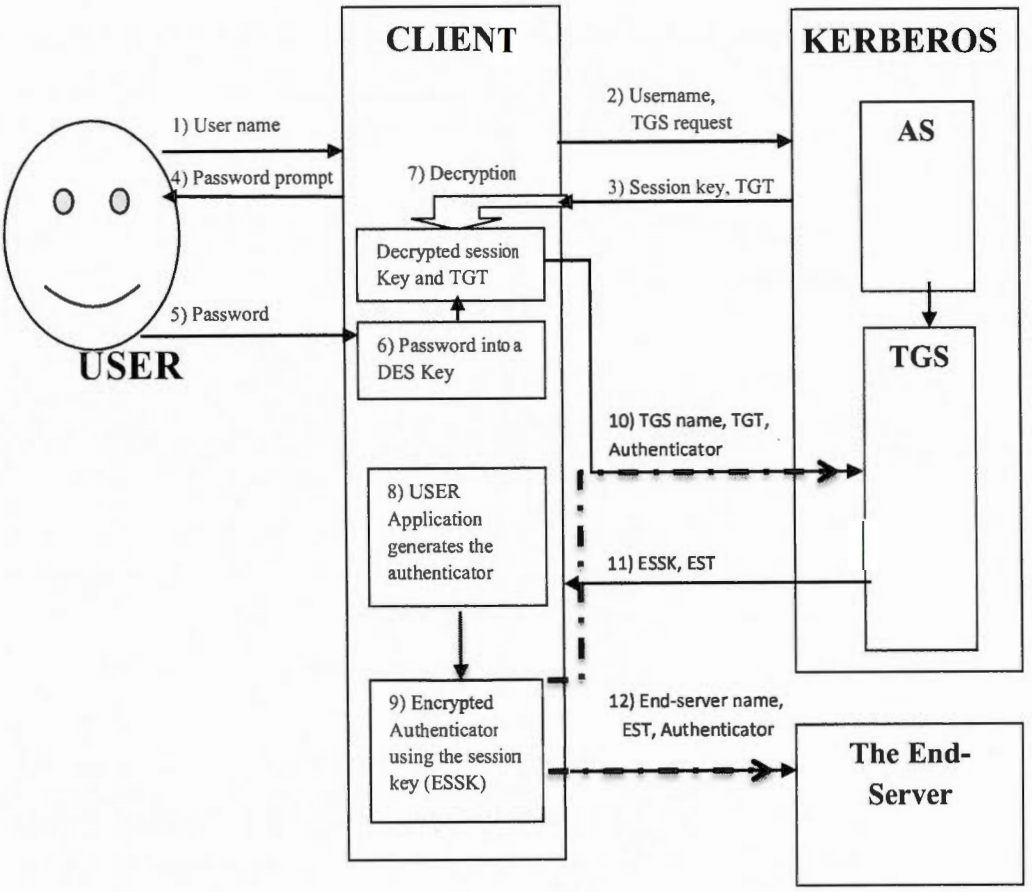

Figure 4.4 The Kerberos Authentication Process

The steps are as follows:

- The user obtains credentials to be used to request access to a service:
  When the user needs to access the network for the first time, he enters his username (1) and the AS gets it from the client along with a request to use TGS (2). The AS checks (in the database) if it has this username, then the AS generates a session key to be sent back to the client[41] along with a TGT (3) that would allow that user to use

[41] All communications between the AS and the client is encrypted with the client private key, which is a key derived from the user password and is known only to both of them. This fact has an exception,

the Ticket Granting Server (TGS)[42]. The session key is used to encrypt the communication between both the client and the TGS. Then the user is prompted for his password (4, 5) that will be converted into a DES key (6). That DES key is to be used by the client to decrypt the AS response (7) that included the TGT ticket as well as the session key, then the DES key and the password will be erased from the client memory and the session key as well as the TGT will be stored for future communications.

- The user requests an authentication for a specific service:
  Once a client is issued a ticket to use the desired server, including the TGS server, the application[43] generates an authenticator (8) including the remote user IP address and the current time.[44] Then the client encrypts the authenticator using the session key (used between the Client and that specific server) (9), and sends the authenticator along with the ticket and the desired server's name to the server (10). This process involving an authenticator created by the application, is called "Service Access protocol", and allows the end user to access a service provided by an organization inner server. The server decrypts both the ticket provided by the client as well as the authenticator, compares their information, and if everything is matching, allows the request to proceed[45]. By the end of this process, the server is assured of the remote user identity.

- The user presents the credentials to the end server:

represented by number 3 in the figure, where the communication is encrypted with a key known only to the TGS and the AS. That's why the user has to decrypt this information, as shown in the figure.

[42] TGS generates access tickets through which the remote user can access the end server. Communication between AS and TGS are all encrypted using a key known only to TGS and AS.

[43] The application is an interface that includes routines and generates private keys (as explained before).

[44] The current time would help identify message replay.

[45] Sometimes the client asks for the server to get authenticated as well, additional steps will then be taken by the AS in order to perform such a request.

As already explained, when a remote user requests a service for the first time, that request has to go to the TGS. Since granting a ticket for the end-server through the TGS is a service by itself, it uses the Service Access protocol just described above, that is also the case for requesting a service from any other server. The TGS checks the authenticator and TGT, and if valid, the TGS generates a new session key to be used between the client and the end-server (End-Server Session Key (ESSK)). Then the TGS builds a ticket permitting the client to use the end-server (End-Server Ticket (EST)), whose life time is the minimum of either: the default time to use that specific service, or the remaining lifetime of the TGT already issued for that user. The TGS sends the EST, along with the ESSK back to the client, all encrypted (11) in the session key used between the client and the TGS. The client then would present that ticket to the end-server, along with a new authenticator (generated using the "service access protocol", as described above, however this time: to enable the client access to the End-server), all encrypted using the ESSK (12).

Now that we explained the Kerberos authentication process in details, we can conclude that this process ensures the following security benefits:

1- By the end of the authentication process, the end-server should be certain of the identity that the client claims to have and, if mutual authentication took place, the client also would be certain of the identity of the end-server as well.
2- Since the clocks are synchronized between the client and the end-server it needs to use, once the end server checks the time of the received EST and finds a significant difference between its time and the present time of the end-server itself, the end-server would consider that EST as a replay for another legitimate EST and will discard it immediately.
3- Both the client and the end-server share a session key to encrypt their mutual communications. Nobody else can have that key, which was generated from within the Kerberos system (from the TGS), thus it is authentic to both party. That session key guarantees that messages between both parties are authentic (especially if they're comparably recent enough, as explained in the previous point).

Now that we introduced a number of remote user authentication methods that can be used either within Dynamic ACL to control access into an organization's NAS router, or by themselves to control access to a router. Let's introduce in the following section the way to combine a number of such authentication methods together for the sake of redundancy, and robustness, by creating a practical approach that can prevent intruders' attacks in a real world situation.

4.5 Creating recovery peers for the authentication server

Before implementing authentication servers on a given to network, it is important to create a certain level of redundancy, in order to avoid situations when the server is no longer able to provide the required authentication service, which might lead to the impossibility of accessing the network by all remote users.

In this section, we introduce the concept of creating peers for the AS (authentication) server, altogether with other related concepts. Let's first introduce some aspects about the NAS router's access, in order to better lead the reader to the explanation of the AS recovery concept.

4.5.1 The difference between Exec access and Privileged access

Actually a NAS router can support authentication for two different modes of access:

1- Exec access
2- Priviledged access

"Exec access" mode is a user access mode that allows the user to use the connection without being able to change any of the router configurations. Exec access mode is usually used for remote users connecting through dynamic ACLs with an intention to reach an inner network resource.

The Privileged Access mode allows the connected user to change the router configuration, and is used for users connecting to the network with an intention to administer the router.

In order to reach the Privileged access mode, the user has first to successfully go through the Exec access mode authentication process in order to access the router in the Exec access mode. Once logged in through exec access mode, the user can go through the

privileged mode authentication process in order to access the router in the privileged access mode. Thus any user connection into the router's Command Line Interface (CLI), whether for Exec or privileged access, starts by successfully logging through the Exec access mode, and then the user might or might not use the privileged Exec mode access, according to his needs.

According to Cisco (Cisco IOS Software release 11.0, 2012), the user can Exec access the router using one of the four line access methods:

- Console, which is a physical socket on a router where a cable can be inserted allowing a computer connection to that router in order to access its Command Line Interface (CLI).
- Auxiliary, which is a physical connector on a router allowing a remote terminal or a PC with a terminal emulator, to access the router CLI using an analog modem.
- TTY, which is a standard asynchronous line used for communication with a computer terminal. It is a teletype-printer or a typewriter equipped with an electronic communication channel, and can still be configured on Cisco 2509 router, using Cisco IOS software version 12.2(19) (Cisco IOS Software Releases 11.0, 2012).
- VTY, the Virtual Terminal Line, allowing terminals to access the router CLI.

4.5.2 Method lists

Since Dynamic ACLs security is considered a front-line prevention against illegitimate user access into the organization's NAS router, they are concerned with securing the first access mode the user has to reach in order to get access to any other mode or organization's resource, namely the user Exec access mode. Thus we focus on explaining the concept of securing the Exec access mode, within the study. In particular, we focus on explaining the concept of enforcing security best practices to ensure the elimination of security gaps that might happen during such access. Thus we will enforce the concept of creating recovery peers for a given authentication server protecting this mode, so that, by any means, if the server is not available to perform its security functions, other servers automatically becomes available to provide it. This process involves the use of method lists, which help determine the order in which the peers will take turns providing such security functions. In other words, method lists determine how the authentication, the authorization and the accounting functions will take place securing the user Exec access mode (Cisco IOS Software Release 11.3, 1999).

The authentication method lists can be used to secure the privileged mode as well, however this authentication is beyond the work to be done by the Dynamic ACLs, as this authentication will only take place after the user successfully authenticates for the Exec access mode and reaches it. Since an authenticated user through NAS Dynamic ACLs might need to access the NAS in order to administer it, we will explain the privilege mode authentication as well.

Now that we explained the router access modes, let's introduce the concept of the peer recovery, which will insure that the authentication process takes place for certain, in order to secure the user access into the organization's network.

4.5.3 Peer recovery mechanism

The authentication method lists specify, to the NAS router, one or more types of authentication methods to be performed as well as the sequence in which they will be tried for that performance. Whether the authentication method is through AAA or non-AAA (through the use of enable password versus an AAA server for instance), the named authentication methods should be put in order in the form of a sequential list; according to the priority they will be queried so that they can provide the user authentication process accordingly. Each of these methods refers to one or more security protocol/server to be used in the authentication process, thus each method defines how the Exec mode access will be authenticated. Next, that authentication method list should be applied to the NAS interface where the authentication should take place. A default method list is automatically applied to all interfaces of the NAS router, except when overridden by a named method list.

In case of using the methods list within Dynamic ACLs context, where user Exec access method uses the corresponding line methods (console, auxiliary, VTY or TTY) in order to reach the router's CLI, the authentication method list should be applied to the NAS lines to be secured, rather than the NAS interface, so that the users accessing the NAS router using any of those lines methods would have to go through the authentication process specified by the method list applied for that line.

If one method on the list fails to respond, the process goes to the next authentication method on the list. When a router successfully uses one authentication method on the list, the rest of the methods are ignored. Thus, except for the first method, a designated method is performed in case of the unavailability of the previous authentication method mentioned on the list, the unavailability of a method might be due to the unavailability of the authentication server or the existence of a connection problem preventing to reach it.

If the authentication fails: this means the user has been denied by one of the authentication methods on the list or by the local username Database. At this point no further authentication methods included on the list are attempted, as the authentication method itself functioned as desired (didn't fail, and the authentication server was available) and rejected the user as a result. That's different from the failure of the authentication method due to

inability to reach the authentication server (or the authentication server is not responding back), thus no authentication takes place at all, neither rejecting nor accepting the user, and thus the next method on the list has to take place. If all authentication methods on the list fail, including the last method, the router will end up in a user denial action, since no authentication took place and no more authentication methods exist on the method list.

Thus the result of an authentication method used by the NAS router can be narrowed down to the following:

1- Success: the method was available and the user was authenticated.
2- Fail: the method was available and the user wasn't able to authenticate.
3- Error: the authentication was unavailable as a service because one of the following cases occurred:
   - The first authentication server in the group of the specified method is unreachable and thus the next server in the group is used.
   - The specified method either doesn't exist, or in case of a group of authentication servers, none of the group servers is reachable.

(Cisco IOS Security Configuration Guide, Release 12.2-AAA overview)

The mechanism of the recovery process can be summarized in the following:

The NAS will query all authentication servers according to the designated authentication methods mentioned on the list, respectively, while going through a pattern until the user gets either authenticated or rejected, or until the user session gets terminated. The fact of specifying multiple types of authentication methods within the method list provides redundancy, since these methods will recover for each other's unavailability.

Figure 4-5 will help the reader better visualize the NAS, the AS, the user and the recovery peers relationship; as follows:

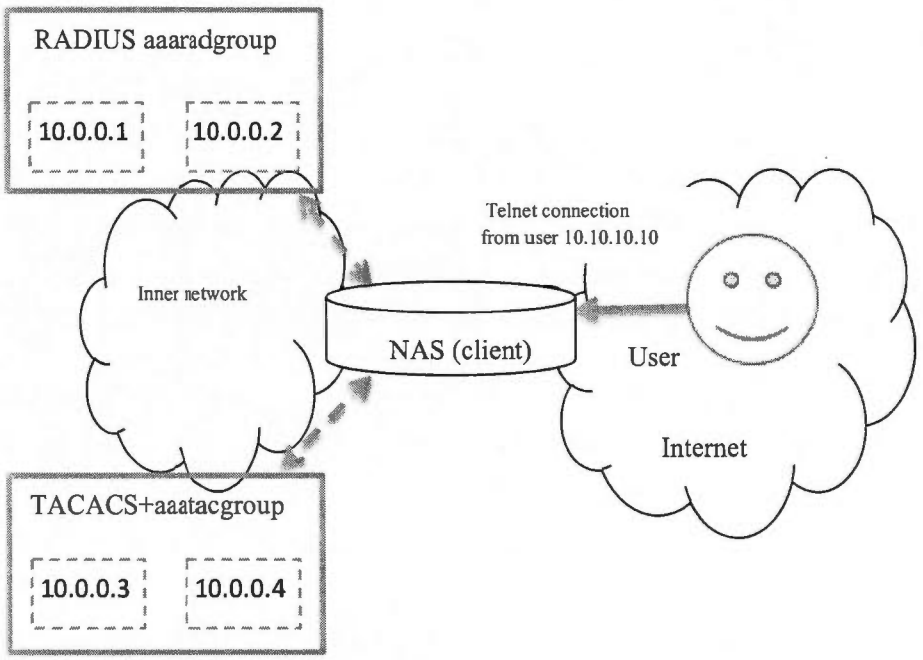

Figure 4.5 The Authentication server (AS)'s peer recovery

As we can see in figure 4-5, the architecture of the network topology might refer to a number of servers that might be planned to be treated as one server group, server group 1, authenticating users through one authentication method, these could be RADIUS servers for example, versus another group of servers, server group 2, authenticating users through a different authentication method, and those latter ones could be TACACS+ servers. In another design, one subset of the first server group, along with another subset of the 2nd server group could be assigned to one authentication method from the method list, while the other subsets of group 1 and server group 2 could be assigned for a different authentication method from the method list.

According to Cisco (Cisco IOS Security Configuration Guide, Release 12.2-AAA overview), each server will be identified within the authentication process, uniquely through

its IP address and a UDP port number. One server can have more than one entry, each identified uniquely in the server group; so that different authentication requests might be sent simultaneously to the same server at the same IP address. Thus, two servers (hosts) belonging to the same server group represent a failure backup for each other, in order to provide the authentication service for the users attempting network access.

If we consider figure 4-5 as an example, it will help a clearer explanation: let's define three authentication methods on the NAS as follows:

- RADIUS server group aaaradgroup, which includes servers 10.0.0.1, and 10.0.0.2.
- TACACS+ server group aaatacgroup, which includes servers 10.0.0.3, and 10.0.0.4.
- Local authentication database using username command

When the user at IP address 10.10.10.10 attempts to access the network, the Cisco IOS configured on the NAS, first tries to contact the first AAA server in the RADIUS aaaradgroup, with IP address 10.0.0.1. If the server is reachable, it will perform the authentication and none of the other methods on the list is needed. In this case, the user 10.10.10.10 has to enter the username/password combination as they're configured on the authentication server, or the user would fail the authentication and will be denied access, and still none of the following authentication methods are tried, since the authentication has been performed already by 10.0.0.1 denying the user.

If the router is unable to reach 10.0.0.1, it will still use the aaaradgroup, contacting the second server 10.0.0.2. If that server is unreachable, the NAS tries the next method on the list contacting the servers in the TACACS+ aaatacgroup, one by one the same way as aaaradgroup servers. If still unable to reach any of them, then the router will try the username authentication method, where the user has to enter the username/password combination that's configured locally on the router, in order for the NAS to verify the user logins. In case neither username command, enable secret command, nor line password was configured on the NAS, when the router attempts to use any of these methods, it will consider them as unsuccessful and will try the next designated method on the list. And since no more designated methods exist on the list, then the user will be denied access.

This example represents an introduction to the concept of peer recovery implementation, and will help the reader to understand the more detailed example presented in the following section.

4.5.4 Peer recovery implementation and configuration

In this section we demonstrate the possibility of combining many types of authentication methods into method lists in order to secure remote Exec user access to the NAS. Some of the combined authentication methods will implement the AAA security services, through the use of AAA servers, while other methods will implement non-AAA services using regular passwords that can be locally configured on the router (like enable password and login password). In order to create such hybrid combination, we have to enable the AAA services on the router before defining the method lists, so that the router would be configured to use such services.

Actually the AAA security services, once enabled on a router, support a variety of login authentication methods. The network engineer would decide to choose a certain number of such authentication methods to recover for each other while performing the authentication process. This is achieved by creating one or more lists of authentication methods that are tried at the user Exec access. Authenticating the user in the Exec access mode to reach the router CLI is referred to as "login authentication", which will be explained in this section.

For user Exec shell access, the syntax commands sown in figure 4-6 must be configured on the NAS:

```
Router(config)# aaa new-model                                    (A)
```

```
Router(config)# aaa authentication login {default | list_name}
method1 [method2...]                                             (B)
```

```
Router(config)# line [aux | console | tty | vty] start_line_#
          [end_line_#]
Router(config-line)# login authentication {default | list_name}
Router(config-line)# timeout login response seconds              (C)
```

Figure 4.6 Peer recovery syntax code

The command in part (A) of figure 4-6 is to enable AAA globally, on all interfaces of the NAS router. This command should be executed before configuring any other AAA command.

The command in part (B) of the figure is to secure Exec access to the router by creating a local authentication method list. The login keyword is to force authentication at login, as it is the specified type of user Exec access method used to reach the NAS line, and is followed by the reference to the authentication method(s) to be used. This reference is a character string specifying the unique name of the authentication method list to be used, followed by the authentication methods included within that list, listed in the order the NAS will perform them. Thus the method argument/keyword refers to the actual method the authentication algorithm tries; a table including such keywords along with their description will follow. A method list can include up to four authentication methods to be tried by the router. An authentication method is tried only if the previous method returns an error, not a failure. In order to configure the router to accept the user authentication even if all methods

return errors, the router can be configured with "**none**" keyword, as a last method on the method list, which is never recommended, according to E-Tutorials (E-tutorials - Authentication).

The default list is used when a method list is not specified in the AAA authentication login command, and it is automatically applied to all router interfaces. To create it, the "default" keyword is used, followed by the authentication methods to be used as default.

Part (C) of the figure corresponds to entering into the line configuration mode as chosen by the network engineer (AUX, Console, TTY or VTY), where the authentication method list will be applied. According to E-Tutorials (E-tutorials - Authentication),The timeout command is optional and it is to specify the time the IOS waits for the user login information before the authentication is timed out. The default timeout is 30 seconds, though it can be set from 1 to 300 seconds. The default number of login attempts permitted by Cisco IOS is three, in order for the user to enter his correct login information. Once the user reaches 3 times, the IOS disconnects him and forces him to repeat his login attempts again; however with AAA, that number can be changed from 1 to 25 attempts. The following command can be added to the router, as an optional configuration to change the number of user login attempts, using AAA servers:

```
Router(config)# aaa authentication attempts login #_of_attempts
```

According to Cisco (Cisco IOS Security Configuration Guide, Release 12.2- Configuring Authentication, 2006), the AAA authentication methods for user Exec access (line login) are summarized in table 4-1, as follows:

Method Keyword	Description
enable	The enable password or the enable secret is used for the authentication. An enable password has to be defined/configured before such an authentication method is used.
Krb5	The Kerberos5 server is used for the authentication. The user password is never sent to the NAS: Once the user is prompted for his username, the Key Distribution Center (KDC) checks for an entry for that user, then creates an encrypted Ticket Granting Ticket (TGT) with that user existing password and sends it back to the router. Then the user is prompted for his password, that the router will use to decrypt the TGT. If that decryption is successful, the user becomes authenticated and his TGT is stored on the router's user credentials cache. Before using the Kerberos authentication method, communication between the router and the Kerberos security server should be enabled.
Krb5-telnet	The Kerberos5 Telnet authentication protocol for authentication, when Telnet is used to connect to the router. If used, this method must be listed first on the method list.
line	The line password, on the line the user is attempting to access, is used for the authentication. A line password has to be defined/configured before such an authentication method is used.
local	The local username, within the router database, is used for the authentication.
Local-case	The case sensitive local username, within the router database, is used for the authentication.
None	No authentication is performed.
Group RADIUS	All configured RADIUS servers perform the authentication. Before using the group RADIUS authentication method, communication between the router and the RADIUS security server should be enabled.
Group TACACS+	All configured TACACS+ servers perform the authentication. Before using group TACACS+ authentication method, the communication between the router and the TACACS+ security server should be enabled.
Group Group-name	Only a subset of RADIUS or TACACS+ servers perform the authentication as defined by the AAA group server radius or AAA group server TACACS+ command. First the group's group name and server members have to be defined by the command: AAA group server. Then the '**group** group-name' should be specified as the authentication method. Before using group name authentication method, the communication between the router and the RADIUS or TACACS+ security server should be enabled.

Table 4.1 The AAA authentication methods for user Exec access (line login)

Now that we explained the use of the authentication method lists to authenticate user Exec access mode, let's demonstrate, in this section, the use of the authentication method lists in order to authenticate the privileged Exec mode access (enable authentication).

In order for the user to access the privilege Exec mode, he has first to be authenticated to access the user Exec mode. Thus when it is time for building the authentication method lists that will determine whether the user is authenticated to access the privilege access mode, the router will be already configured with the authentication for the Exec mode which includes the AAA authentication services enabled, as shown in the code above, thus there will be no need to reconfigure that feature on the NAS router again.

Building the authentication method lists for the privileged access mode then will take place right away; though unlike the User Exec mode, the Privilege Exec mode doesn't allow building many authentication lists, only the "default" list is allowed to be built. This makes sense: because there is only one-way to access the privilege access mode for the remote user: through the user Exec mode, thus after going through the users Exec mode authentication process successfully. Also in user Exec mode, there exist many different types of login access methods/lines (console, auxiliary (AUX), TTY, VTY), while in the privilege mode, the user accesses the NAS from within, as he would have already accessed the user Exec mode.

According to Cisco (Cisco IOS Security Configuration Guide, Release 12.2- Configuring Authentication, 2006), for user privilege shell access authentication method list, the following syntax commands must be used:

```
Router(config)# aaa authentication enable default method1
[method2...]
```

Figure 4.7 Privilege access authentication method list – Syntax code

Thus the command is to enable login checking for the users accessing the router through the user privilege Exec mode. The authentication methods allowed to be used within the default authentication method list are the same as the methods allowed within the user

Exec mode except for "local" and "local-case" authentication methods. The privilege access mode authentication will be elaborated in an example that will follow in the next section.

In order to better understand the mechanism for AAA authentication methods, let's elaborate the algorithm that the NAS IOS follows in order to secure user access in general, as follows:

1. Whenever a remote user attempts to login into the NAS, The NAS checks if the AAA services are available.
2. The NAS recognizes the authentication method list that is configured along with that specific type of user access, and recognizes the authentication methods specified within that list.
3. The NAS proceeds with the first method on the list, and after the authentication takes place, the NAS receives an authentication message (including the authentication result) from the authentication server performing the authentication.
4. If the authentication message is an error, then go to step 5; otherwise: if it is a fail (deny), then no access is allowed, and the authentication process stops. If it is a success (pass), then the user is allowed access, and the authentication process stops.
5. The NAS proceeds with the next authentication method on the list, then it will go to step

The following configuration example of figure 4-8 will help understand the way the NAS can be configured with a number of authentication method lists in order to secure many types of router access methods. The configuration is followed by an explanation about how the router processes such authentication methods.

```
Router(config)# aaa new-model                                    (1)
```

```
NAS(config)# tacacs-server host 10.0.0.5 single-connection (2)
             key secret5
NAS(config)# tacacs-server host 10.0.0.6 single-connection
             key secret6
NAS(config)# tacacs-server host 10.0.0.7 single-connection
             key secret7
```

```
NAS(config)# aaa group server tacacs aaatacgroup              (3)
NAS(config-sg)# server 10.0.0.5
NAS(config-sg)# server 10.0.0.6
```

```
NAS(config)# aaa group server radius aaaradgroup              (4)
NAS(config-sg)# server 10.0.0.1
NAS(config-sg)# server 10.0.0.2
NAS(config-sg)# server 10.0.0.3
```

```
NAS(config)# aaa authentication console group aaatacgroup (5)
local

NAS(config)# username admin1 secret cisco1                (6)
NAS(config)# username admin2 secret cisco2

NAS(config)# aaa authentication login default             (7)
group aaatacgroup group tacacs+ group aaaradgroup krb5

NAS(config)# aaa authentication enable default            (8)
group aaatacgroup enable

NAS(config)# enable secret OutKeep                        (9)

NAS(config)# aaa authentication attempts login 1          (10)

NAS(config)# line console 0                               (11)
NAS(config-line)# login authentication console

NAS(config)# line vty 0 15                                (12)
NAS(config-line)# login authentication default
```

Figure 4.8 Example of NAS configuration using authentication method lists

The configuration code of figure 4-8 demonstrates the different types of authentication methods to be applied to different types of access to the router. These access types covers both modes: the User Exec mode as well as the privilege mode. Also the configuration code demonstrates the use of authentication server groupings, and how to specify a certain set of a given server grouping versus a whole group of servers.

The example starts by enabling the AAA functions on the NAS router, then specifying the TACACAS+ server grouping and subgrouping, as well as the RADIUS server grouping in command sets 1 to 3.

Next, the configuration code defines the authentication method lists to be followed with console access, user Exec access (login) in command sets 5 and 7, respectively. Each method list identifies the authentication methods to be followed in a certain order, where every method has to be tried by the NAS router in case the previous method was not available.

Next, the configuration code defines the authentication method lists to be followed when the user wants to privilege access the router from within the Exec access mode, in order to reach the Privileged mode access (Enable), as shown in command set 8.

Then, the configuration code applies the specified authentication method lists for each access Exec method mentioned (console and VTY), in command sets 11 and 12 respectively as specified by the following explanation.

Before we proceed with the code explanation, it is worthwhile to clarify that the default authentication method list has to be used by the router's IOS whenever there is a type of login access without an authentication method list specified. E.g., if we consider figure 4-8's configuration, there is no specification for the authentication method list to be used for the auxiliary line access (AUX), since this type of VTY lines are not mentioned in the configuration, hence, lacking all reference for an authentication method list assigned to it, so the NAS IOS will automatically assign it the default authentication method list in order to secure this type of access.

The code of figure 4-8 can be explained as follows:

1. This command enables AAA globally on all NAS interfaces.

2. These three commands specify the three TACACS+ servers used to perform the authentication process (as a grouping). This grouping of servers represents a whole set of TACACS+ servers configured on the router, and can be referenced as a potential authentication method within an authentication method list to be built. Referencing this method would be implicitly by mentioning the word "TACACS+".

3. These three commands set up a subgrouping including only two specified TACACS+ authentication servers. This subgrouping can be referenced as a potential authentication method within an authentication method list to be built. Referencing this method would be explicitly my mentioning the subgrouping name: "aaatacgroup".

4. These four commands set up a grouping of three RADIUS authentication servers. This grouping can be referenced as a potential authentication method within an authentication method list to be built. Referencing this method can be done either implicitly by mentioning the word "RADIUS", or explicitly my mentioning the subgrouping name: "aaaradgroup"[46].

5. This command builds an authentication method list for user EXEC access mode. The access authentication method list's name is "console" (to refer to a method list to be used with console access), and it has two authentication methods within its method list:
   - The first authentication method (**group aaatacgroup**) uses only the TACACS+ subgrouping, two servers specified within the aaatacgroup to perform the authentication process. Thus the NAS will first try TACACS+ server 10.0.0.5 to perform the authentication; if it returns an error, then the NAS will try TACACS+ server 10.0.0.6 to perform the authentication; if it returns an error, then the NAS will try the next authentication method on the list, since no more TACACS+

[46] Referencing a method implicitly by mentioning the word "RADIUS" or "TATACS+" implies trying all RADIUS (or TACACS+ servers that are configured on the NAS router, since they all act as recovery peers for each other in this case. However, referring explicitly to a specific subgrouping name, e.g. "aaaradgroup", specifies only the specific servers mentioned in that subgrouping, rather than all other RADIUS servers that might be configured on the NAS router.

servers are configured on the NAS router.

- The second authentication method (**local**) uses the local database username/secret password logins, specified within the **username** commands that are configured globally on the NAS (to involve all NAS interfaces). Since the username local logins are configured on the NAS in part # 6, then the local authentication method can take place. If there were no local logins configured on the NAS, the local authentication method would have returned an error, and since no more authentication methods are listed on the "console" method list, then the user will not be authenticated and will be denied access to the NAS.

  This "*console*" authentication method is referenced in part # 9, where it is applied to the console access line configuration, as the authentication method list to be followed in order to authenticate users during their console login to the NAS. The use of the word "*console*" as a name for the authentication method list is a descriptive term; any other name can be used to describe this authentication method list.

6. These commands define the two logins to be used when users need to access the router directly through the console.
7. This authentication method list includes the following four authentication methods:
   - The first authentication method (**group aaatacgroup**) tries only the two TACACS+ servers specified within the aaatacgroup to perform the authentication process, as it is explained above in step # 5. If both TACACS+ servers return an error, then the NAS will try the next authentication method on the list.
   - The second authentication method (**group TACACS+**) tries all three TACACS+ servers configured on the NAS router to perform the authentication. In this example, there are three servers configured: thus the NAS will first try 10.0.0.5, if an error is returned, The NAS will try 10.0.0.6, if an error is returned, the NAS will try 10.0.0.7, and if an error is returned, the NAS will try the next authentication method on the list, since no more TACACS+ servers are configured on the NAS router.
   - The third authentication method (**group aaaradgroup**) tries all three RADIUS servers specified within the aaaradgroup to perform the authentication process: thus the NAS will first try 10.0.0.1, if an error is returned, The NAS will try 10.0.0.2, if

an error is returned, the NAS will try 10.0.0.3, and if an error is returned, the NAS will try the next authentication method on the list, since no more RADIUS servers are configured on the NAS router.

- The fourth, and last authentication method (**krb5**), tries the Kerberos authentication server to perform the authentication process: if an error is returned, the NAS will have try the next authentication method on the list, since no more authentication methods exist on the list, then the NAS router will deny user access.

8. This command builds an authentication method list for privileged EXEC access mode, using the "**enable**" access method. The access authentication method list's name is "default" (which is a keyword referring to a method list to be used by default in case no other lists are specified), and it has two authentication methods. If both methods return error as authentication results, then the user will be denied enable/privileged access to the router.

   This authentication method list includes the following authentication methods:

   - The first authentication method (**group aaatacgroup**) tries only the two TACACS+ servers specified within the aaatacgroup to perform the authentication process, as it is explained above in step # 5. If both TACACS+ servers return an error, then the NAS will try the next authentication method on the list.
   - The second authentication method (**enable)** uses the enable secret password logins, mentioned in command # 9 and configured globally on the NAS (to involve all NAS interfaces). Since the enable secret[47] is configured on the NAS, then the enable authentication method can take place. If there were no enable secret configured on the NAS, the enable authentication method would have returned an error, and since no more authentication methods are listed on the "default " method list to secure the privilege access, then the user will not be authenticated and will be denied access to the NAS.

9. This command identifies the enable secret password logins, configured globally on the NAS, thus applied to all NAS interfaces.

---

[47] Enable secret is the password configured on the NAS for user Privileged access mode (Enable mode).

10. This command is to restrict the number of user authentication attempts, during a login session. The number specified is one, implying that in case the user cannot successfully authenticate on his first trial, his session will be terminated, and he will have to reestablish a new access session to NAS in order to retry to authenticate.

11. This command applies the "*console*" authentication method list, which is the authentication method list mentioned in part # 5, to the console access line configuration, specifying how authentication should be done trying both authentication methods that are included, mentioned above, to secure user access through the console line.

12. This command applies the login "*default*" authentication method list, which is the authentication method list mentioned in part # 7, to the Virtual Terminal (VTY) access line configuration, specifying how authentication should be done, trying all four authentication methods that are included as mentioned above, to secure user access through the VTY line.

This chapter has studied the AAA paradigm, the authentication servers/protocols as well as the peer recovery concepts in depth, through introducing a thorough explanation as well as a deep theoretical analysis of each concept, presenting the mechanisms related to the details introduced, while developing configuration codes that help understand and emphasize the different approaches, including different configuration options that are related to their application.

In the chapter, we introduced the different remote user authentication methods to secure user access into the NAS router. We started by defining authentication as a security concept in order to emphasize its importance, along with the AAA paradigm as a concept that dictates the use of authentication servers/protocols. Next, we introduced the mechanism of an authentication server, followed by the different secure access approaches that are usually used along with the authentication servers, like Challenge/Response, OTP, PPP, PAP, CHAP and proxy servers. Then we introduced the different authentication servers/protocols in details, analyzing their different mechanisms while concentrating on the authentication functions for each of them. Thus we introduced RADIUS, DIAMETER, TACACS+ and Kerberos. The latter was thoroughly analyzed to justify the different nature of its mechanism, which prevents us from further considering it in this study for the

authentication comparison using Dynamic ACLs concept. Finally, we introduced the concept of the AAA authentication method lists that are used to create a transparent peer recovery for the different user login access methods as well as with the enable method. We proposed a syntax code as well as a simplified example and a full NAS router configuration coded example that demonstrate different configurations of TACACS+ and RADIUS using different user login methods and as well as enable methods. This code can be considered as a reference to understand and apply the AAA server authentication for different user access methods to any organization's NAS.

This chapter's importance stems from its thorough explanation for the concepts mentioned above, ending by relating most of such concepts into a well developed code that put them all together into a practical application, and that will help us explain/develop further more sophisticated concepts and configurations codes about dynamic ACLs authentication using the AAA servers, as will be introduced in chapter 5.

In the next chapter, we explain the authentication process that takes place within the Dynamic ACLs as a security technique. And based on the analyses of the concepts introduced in chapter 4, we discuss the possibility of the application of the authentication concepts explained within the dynamic ACL's authentication process, along with the comparison between them, which will allow the reader to better understand their weaknesses and strengths, and to better make his own choices either by applying them as a complementary option or as a substitute for the dynamic ACL process.

CHAPTER V

AUTHENTICATION AND DYNAMIC ACLS

The study's main focus is the elaboration of the user authentication step, which is the main step upon which relies the Dynamic ACLs filtering process and which traditionally involves the use of Telnet as a virtual line connection, as a way to access the NAS upon which the filtering process takes place.

Dynamic ACLs, also known as Lock-and-Key, are used to authenticate a remote user, by opening a temporary hole in the extended ACL that filters the user access to the organization inner resources. Lock-and-Key is typically used in small networks to authenticate a specific type of remote users' access[48].

In this chapter, we will analyze Telnet as a connection method, as well as an authentication mechanism, and its drawbacks that constitute some security limitations, and sometimes security risks, while performing remote user authentication within dynamic ACLs filtering process. Then we will introduce the use of other connection and authentication methods and how they can help overcome such drawbacks, when authenticating dynamic ACLs' remote users.

Thus we will introduce some combination case scenarios of different authentication methods in order to compare them in terms of scalability, reliability, ease of configuration, manageability, and cost efficiency.

Finally we will introduce some application recommendations built upon the research conclusions obtained from such comparison, in order to provide the study's recommendations.

---

[48] Lock-and-key is to be configured on the NAS router when it doesn't have the Cisco IOS firewall feature set with authentication proxy installed, as authentication proxy is considered as a substitute for Lock-and-key, as we will elaborate within this chapter.

5.1 Dynamic ACLs and choosing an authentication method

Before we analyze the Telnet authentication mechanism, let's recall that during dynamic ACLs, the NAS router has to authenticate the user connection, and can be configured with three choices of authentication methods:

- Using the local database through **username** command.
- Using the VTY line through **password** command.
- Using an AAA server.

Out of the three authentication methods, let's consider the AAA server implementation, which is a little different than the other two authentication methods. There exist two case scenarios for the implementation of the distributed application of the AAA servers as a dynamic ACL's authentication method. Implementing such an authentication in a networked environment can be done through one of two approaches:

1- In the first approach, the authentication server is located on the user's network. Performing such an approach will require the server to be located on every remote user boundary firewall, within the user's premises. Such an application approach of AAA authentication is impractical, extremely expensive and, thus, inapplicable.

2- In the second approach, the authentication server is located within the organization network, in order to avoid the previous infeasible, high priced authentication method. With this second approach, the authentication takes place within the organization network premises and thus, can be applied as a distributed authentication process. As a conclusion, the second approach is the way AAA servers are implemented in real world networks.

Thus in order to reach into the organization network, the user has first to reach the organization NAS using Telnet as a VTY session, allowing the remote user authentication process to take place, according to the configuration of the local router, even though this connection is theoretically not secure enough to transmit the user's credentials reaching AAA authentication server.

In case the user fails to get authenticated, he will be re-prompted to re-enter his authentication information[49]. Successful or not, the authentication process as well as its corresponding VTY connection will be terminated by the router. Thus the only purpose of Telnet is for authenticating the remote user, so, it will no longer be needed after the user successfully becomes authenticated, since the user will be dynamically granted access for the organization's inner resources through the Dynamic ACL.

Before we introduce the Telnet mechanism in details, let's elaborate the reason why Telnet is often used with dynamic ACLs authentication process, at least as a first step for connecting remote users to reach the organization NAS.

As we explained through "the Dynamic ACLs Configuration" section of Chapter 3, Dynamic ACLs have to run an autocommand in order to execute the dynamic entry entered into the extended ACL. Such an autocommand has to run within a shell. Therefore, the connection performed by the user into the organization NAS (border router), as the very first step of the dynamic ACLs occurrence, has to be a connection that allows a shell environment where the autocommand can be run, such as Telnet.

However, Telnet is not the only connection method that would allow user access to the NAS Command-Line Interface (CLI), through a shell connection. According to Odom (Odom, 2009), accessing a Router's CLI can be done by one of the three possible methods:

1- Console access,
2- Telnet access,
3- Secure Shell (SSH) access.

Out of the three connection methods mentioned above, only Telnet and SSH can be established remotely, thus we will study them in the following section, in order to compare their security aspects.

[49] The number of re-prompting trials is to be configured depending on the organization's security policies.

Before we get to that comparison, let's introduce the dynamic ACLs mechanism using a Telnet connection, established between the remote user and the NAS.

5.2 Dynamic ACLs mechanism using Telnet

As explained before, in order for a Dynamic ACL user authentication process to take place through the NAS, the NAS should be configured with Lock-and-Key/Dynamic ACL[50], and a Telnet connection has to be established between the user premises and the NAS.

During such a Telnet connection, the user has to pass the authentication process before access is allowed through the network resources. Once authenticated, the user Telnet session terminates, and the NAS creates a temporary entry in the dynamic ACL to allow user temporary access for a certain range of network resources. Such a range can be limited by a wise configuration of the temporary entry, in a way that complies to the security policies of the organization.[51] Then user exchanges data with the inner resources he's granted access to, by the dynamic ACL entry configuration. Such resources would otherwise be denied without the dynamic ACL entry. Once the configured timeout (either the idle or the absolute) is reached, the temporary entry is deleted by the NAS. The temporary entry can also be deleted once the network engineer manually deletes it out of the dynamic ACL's configuration.

If the authentication process fails, the user will only be granted access to the inner resources specified in the static ACL configured on the NAS, without a dynamic ACL entry. Thus the dynamic ACL entry is triggered by the success of the user authentication process.

Now that we explained the dynamic ACLs' mechanism using Telnet as a VTY connection, let's introduce the problem of our research, to guide the reader for a better grasping of the objectives of this study, as follows:

---

[50] Lock-and-Key/Dynamic ACL is a static ACL that can accept a dynamic ACL entry.

[51] The temporary entry drawback is that it doesn't allow flexibility when permitting access to inner resources, in order to personalize such permissions to different privileges specified to different user profiles. As you will see later in the chapter, you cannot set up per-user access policies. Instead, you define one policy for all your lock-and-key users, and this single policy is applied to all the authenticated users.

5.3 Problem of the research

The research focuses on studying Dynamic ACLs, when they're used to restrict remote user access to an organization's inner network[52]. In particular, the study's targets the user authentication step, which is the first step of the Dynamic ACLs filtering process and which involves the use of Telnet connection to carry the user's credentials before a given authentication process begins. Exposing user logins during a certain connection can be the opened door for many malicious attackers to get access through such a connection. Specially that the Dynamic ACL access allows Telnet, as an external event, to place an opening into the NAS/ border firewall, which even increase that router security risks.

In fact, the user authentication process is built, in the first place, upon trusting the identity of the user telnetting into the network border router (NAS): During the Telnet session the user has to enter some login information so that the authentication process begins. Thus establishing a user Telnet session is considered, as an essential connection process to access the NAS, before getting the NAS to start any further authentication for the user.

The study's main focus is to elaborate the security aspects of the Telnet process as a VTY connection to access the NAS, on one hand, and as one of the choices of Dynamic ACLs user authentication methods configured on the NAS, on the other hand.

Thus, we will analyze Telnet security drawbacks that can affect the dynamic ACLs' user authentication process. Other VTY connection methods that can be used in lieu of Telnet as a first step to access the NAS dynamic ACLs, namely SSH, will be explained and compared to Telnet. Then we will compare the different authentication methods involved in Dynamic ACLs user authentication process, in terms of security aspects, scalability, reliability, ease of configuration, manageability, and cost efficiency; in order to create one or more scenario(s) as guidelines that can enable network engineers to choose a more secure user authentication process to be used within dynamic ACLs' filtering process.

---

[52] Dynamic ACLs can also be used to restrict a group of inner users' access to a host on a remote network protected by a firewall. This use of Dynamic ACLs is out of the scope of our study.

Now that we pinpointed the main focus of our study, let's begin by explaining Telnet as a VTY connection method to reach the NAS command line interface (CLI) on the organization border, as well as SSH that can substitute Telnet in this matter. Let's compare SSH with Telnet, in order to decide which option can be considered as a more secure VTY connection method. Thus in this section, we will explain each connection method in details, so that we can thoroughly compare between them in terms of different comparison criteria.

5.4 Telnet as a VTY connection

Telnet is the standard terminal-emulation application layer protocol in the TCP/IP protocol stack. Telnet is used for remote terminal connection, enabling users to log in to remote systems command line interface (CLI) and to use recourses as if they were connected to a local system.

Using Telnet, users connect to the remote systems (devices, servers or company resources), which are called "Telnet servers", using a terminal emulator, which is called a Telnet Client, through an IP network connection. Telnet client applications are available for most computer platforms. Port 23 is the connection port used by Telnet on the Telnet Server (the Router or NAS, in this case) in order to allow user connections into the Server's CLI.

Figure 5-1 will help better visualize the NAS, the AS and the user relationship; as follows:

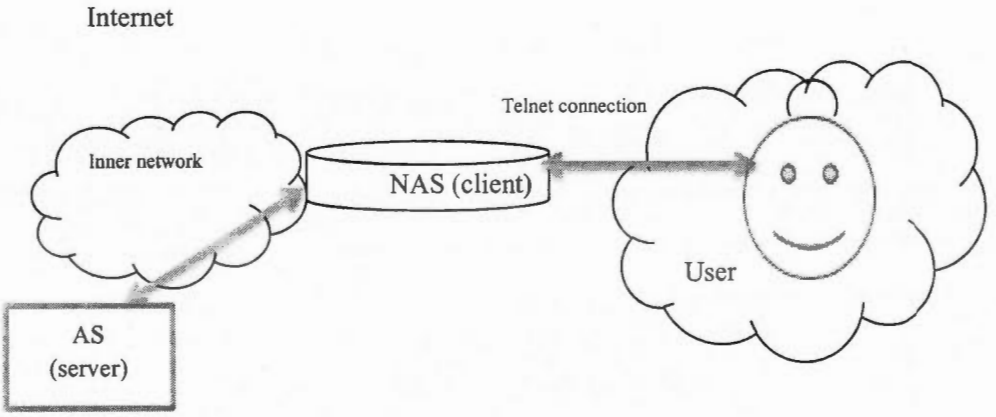

Figure 5.1 The Telnet connection

5.4.1 Telnet vulnerability

A connection that lacks reliable user authentication is considered as an unsecure connection. In this section we will explain the main security risks associated with the use of Telnet as a connection method, the limitations of user authentication security.

- Telnet 's main security issue is that it sends user login information across the network in Plain text, where the username/password will be unencrypted and visible throughout the network), thus exposing the network to many security attacks.

  Telnet sends all connection data as a clear-text data type, thus the login password is exposed, all the time, for potential security risks. Using a packet analyzer, a network sniffer that can intercept network traffic, decoding its raw data and its field values, an intruder can obtain the user credentials or even the message contents from any Telnet communication (Odom, 2009).
  Thus we can get to the conclusion that Telnet authentication process is not secure.

- By default, Cisco Routers only allow console access without a password, and prevent all remote access (including Telnet and SSH) whenever there is no password configured on the router within these access modes. Thus if a VTY line password, an enable password, or an enable secret is not configured on a Cisco Router, there is no way for a remote user to access it through Telnet or SSH[53], which insures some level of security about both Telnet and SSH. Thus in order for the router to be reached, it has to be configured with an enable password as well as line usernames/passwords. However, accessing the router using Telnet doesn't require the user to enter a combination of username/password, only a password is needed, which proves Telnet vulnerability, specially when compare to SSH, which requires a username/Password combination (Odom, 2009).

[53] User can connect to routers without a VTY password in case VTY is configured with "**no login**" command, which disables all authentication checking and thus, would allow all users to directly access a router's CLI, which is never recommended for any local or remote connection, from a security point of view.

- This will lead us to the next risk of using Telnet as exactly stated by Wikipedia (Wikipedia - Telnet Security, 2012): "Most implementations of Telnet have no authentication that would ensure communication is carried out between the two desired hosts and not intercepted in the middle." Thus, throughout a Telnet connection, intruders might interpret the message contents, during their end-to-end transmission between the user and the authentication server. This risk of revealing the connection contents is known as a man-in-the-middle attack, as explained in chapter 1.
- A third risk is about Telnet daemons, which are the programs running without a user direct control or interaction and are proven over time to have some vulnerability problems, which represent potential possibilities of malicious attacks (Wikipedia - Telnet, 2012).

Now that we introduced a number of Telnet security risks, let's consider some solutions to face them.

5.4.2 Overcoming Telnet's drawbacks

Overcoming Telnet security issues can be achieved by following some security procedures. Some of them will be explained in this section, however interested readers can find some more in Appendix A of the study, entitled: "How to improve Telnet security issues?" The procedures that the study recommends to help diminish Telnet security vulnerability, by integrating alternative options, as follows:

- Through using extended Access lists, applied inbound on each and every interface of the NAS router. This suggested solution, though, is not very practical or feasible due to the large number of interfaces that might exist on such routers.

- Through controlling VTY line (Telnet or SSH) access by applying the well-known "Access class" within the VTY line of the NAS router. However, this solution depends

on limiting the source addresses telnetting to the NAS into a defined list or subnets (Odom, 2009). The configuration code of this solution is as follows:

```
1 NAS (config) # access-list 10 permit 10.100.50.0 0.0.0.225

2 NAS (config) # line vty 0 15
  NAS (config-line) # access-class 10 in
```

Figure 5.2 The Telnet configuration using "access class"

According to the configuration code, the access list mentioned in part 1 limits access exclusively to the IP addresses that range from 10.100.50.0 to 10.100.50.225; while part 2 corresponds to applying that access limitation to the VTY line.

Limiting the source IP addresses accessing to the VTY line, as a solution, doesn't prevent the source address spoofing problem and the other potential login exposure, associated with the use of Telnet. Besides, this solution doesn't work with the regular application of Dynamic ACLs that heavily depends on the ever-changing remote source IP addresses (due to DHCP or due to the remote user changing physical positions) that would like to connect to the NAS router.

- In order to overcome the possibility of spoofing, encryption can be configured, so that all traffic (including the source IP address) coming from remote users would be encrypted at a secured remote router and decrypted locally at the serial NAS interface providing the Lock-and-Key. Since, this way, all dynamic ACL traffic entering the NAS would be secured; no hackers would be able to spoof the source address, as it would be impossible for them to duplicate the encryption, in order to be successfully authenticated, as it is a required first step to trigger the dynamic ACL entry, and to reach the inner network resources. This solution transforms the Telnet process into one that is equivalent to SSH's.

- In order to face the password exposure security risk of Telnet, the following steps can be followed:

1. The NAS can be configured with "service password-encryption" command, which can be configured at any given time. According to Odom (Odom, 2009), the service password-encryption is a feature that helps encrypt all existing and future passwords strings that are already stored on the router as well as passwords newly entered onto the NAS, so that they would not be shown within the router configuration, as a clear text.
   Although the password encryption is not extremely stronger than the normal "enable password command", it might prevent casual observers from seeing Telnet passwords in the clear.

2. The NAS can be configured with "enable secret" command, which uses a stronger hash algorithm than the normal "enable password command", as it depends on applying the Message Digest 5 (MD5) mathematical function to the password strings.
   According to Odom (Odom, 2009), the enable secret help defining a certain password to be entered by the user once he would like to privilege access the NAS router, however it can never be shown within the router configuration, as a clear text, which is a detail unavailable in regular passwords.

By now, we can conclude that Telnet has a considerable number of security issues, therefore, we don't recommend its use as connection method for remote users: thus, restricting telnetting altogether, while finding a substitution connection method, is one way to avoid its security disadvantages. **Therefore, in this study, we will elaborate SSH as another more secure VTY connection method that can be a substitution for Telnet[54].**

[54] A VPN connection can be considered as another option to substitute a Telnet connection.

5.4.3 SSH as a Telnet substitute

Actually, according to Wikipedia, because of the security issues with Telnet, its use for many connection purposes has waned in favor of SSH. Thus, let's ask the following question: Why is SSH considered as a more secure authentication than Telnet?

In this section, we will compare both Telnet and SSH, according to their functionality and their security aspects.

Both Telnet and SSH are very similar as they both are characterized by the following features:

1. Both are CLI (shell) connection authentication methods,
2. Both require an IP network connection to access the CLI USER mode,
3. Both require a Client software with a terminal emulator to be installed on the user host,
4. Both use TCP, with a well known port on the Server (Port#23 for Telnet, and Port#22 for SSH),
5. Both have the server receive the text messages, processes them in terms of commands and resend the results to the client.

SSH, however, requires both a username and a password from connecting users, while Telnet doesn't require any. So the NAS router must be configured to use one of the two user authentication methods that require both a username and password: one method with the username and password configured locally on the switch database, or the other method with the username and password configured externally on an AAA server.

Also SSH uses encryption for all data exchange between the SSH client and SSH server (login data and message contents), which provides a more secure connection than Telnet, by protecting user login information and data from being intercepted by intruders.

Also SSH uses public key authentication in order to authenticate the interconnected hosts and to assure they're the intended recipients within a certain communication. In fact, SSH clients use SSH protocol to connect to SSH authentication server, which would be the

NAS in Dynamic ACL application case. However, in case of SSH, the NAS (cisco IOS image) has to support RSA, which is an encryption algorithm that generates public and private key pairs; Data Encryption Standard (DES) or 3DES, as well as password authentication. Cisco Internetwork Operating System (IOS) version 12.1(3) is recommended to be installed on the NAS to guarantee a client sever SSH connection, also SSH version2, which is the latest version, is recommended to be used for such a connection.

Thus, SSH is considered a more secure VTY connection method, providing more reliable authentication algorithms than Telnet. Setting the VTY access to SSH, even though considered as optional, it is highly recommended.

SSH configuration would look like the following:

```
Router(config)# hostname router_name
Router(config)# ip domain-name DNS_domain_name

Router(config)# crypto key generate rsa

Router(config)# username name secret password

Router(config)# line vty 0 4
Router(config-line)# transport input ssh
Router(config-line) login local
```

The configuration starts by assigning a host name and a domain name to the NAS router Then The NAS IOS will generate a public key that will be sent to the SSH client and a private key that will stay at the server, once the command "crypto key generate RSA" is executed. Then, both the global "username" command as well as the VTY "login local" command are to set up the authentication to be performed locally, by the NAS router itself. AAA server authentication can be applied using SSH connection as well.

Now that we explained both Telnet and SSH VTY connection methods mechanisms, as well as compared their security aspects, and their authentication mechanisms, this section

will demonstrate the different cases where Telnet or SSH, as connection methods, can be combined with other authentication methods, in order to perform the authentication process required by dynamic ACLs mechanism.

5.5 Telnet as an authentication method

As we explained before, the remote user authentication through the authentication server method is impractical to be performed at the user premises, thus user authentication has to take place locally, on the organization NAS, or on an authentication server after connecting to the NAS. Thus, in all cases, a user connection to the NAS has to take place first, and such a connection will be done using either telnet or SSH. During such a connection session, the NAS can perform one or more user authentication processes, to guarantee the legitimacy of the user identity, as a main requirement for the dynamic ACL entry to be dynamically placed within the Lock-and-Key ACL, which would grant the user access to the inner network resources. The authentication process, as mentioned above, can be through the use of Telnet as one of the authentication methods that can be configured upon the NAS. In this section, we will analyze Telnet as remote user authentication method, comparing it to other authentication methods that can take place when configuring dynamic ACLs on the organization's NAS. We will also discuss the subject of combining one or more user authentication methods into Dynamic ACLs, to guarantee an even higher level of security for the organization's inner resources.

Tenet as an authentication method takes place under Dynamic ACLs authentication methods, as we will explain in the following section.

5.6 Dynamic ACLs authentication methods

Let's, then explain the three main authentication methods used within VTY line along with Dynamic ACLs' application:

5.6.1 Telnet as a user authentication method

When explaining Telnet as a user authentication method, which is also referred to as "line password", it is highly recommended not to use it within Dynamic ACLs, because the password is configured for the Telnet port on the NAS router, not for the remote user. Therefore every user would have to use the same password, and any user who knows the password will be able to authenticate successfully. Thus it will make no sense to authenticate the user upon such unsecure login information. The configuration syntax is as follows:

NAS (config-line) # **password** *password*
NAS (config-line) # **login**

The first command assigns a password to a terminal or to another device on a line (VTY).

The second command enables password checking at login according to the password specified locally on the router's VTY line, by the previous command.

If applied, and since it is related to the port versus the user, line password authentication would allow all illegitimate users, once they know that common password, which is likely to happen as it is assigned to the port. Having the password, the illegitimate users will easily pass the authentication process, which will trigger the dynamic ACL entry that, in turn, would grant such users legitimate access; exchanging data with the organization secure inner resources. This problem might appear like the one used by any system using a given password, however, the fact that the password is assigned to the Telnet port makes it so widespread to be known for a large number of users, which increase its vulnerability level.

However, there can be one or more preliminary authentication steps for the user to go through, before he actually authenticates using Telnet. Actually, when configuring dynamic ACLs, it is best practice to combine many authentication methods. The following combination of authentication methods along with Telnet might be of a great interest for our research, in order to find a better authentication security while using line password:

Sometimes dynamic ACLs are referred to as "Double authentication", according to E-Tutorials (E-Tutorials - Lock-and-Key Overview, 2012). With double authentication, the dialup user is first authenticated through PPP-CHAP, then through the lock-and-Key ACL.

Double authentication has two stages, according to Cisco (Cisco IOS Security Configuration Guide, Release 12.2-Configuring Authentication, 2006): the first stage authenticating the remote host, and the second stage authenticating the specific users on the host. TACACS+ and other AAA servers can be used to perform both authentication stages, providing a stronger authentication technique than authenticating only the host, leaving the same host password to be used by all host users, as it is the case with the use of line (Telnet) authentication method by itself.

And since PPP-CHAP OSI layer-2 authentication method depends upon a strong encryption mechanism, as explained in Chapter 4 titled "Authentication methods", combining it and configuring it to take place as a first authentication stage before the layer-7 Telnet as a second authentication stage, would provide a more reliable user login, personalized according to each user. Also, CHAP's strong encryption aspect will eliminate the possibility of eavesdropping over the user logins, which exists in case of using Telnet authentication, which allows a chance of overcoming some of Telnet authentication security drawbacks (Cisco IOS Security Configuration Guide, Release 12.2-Configuring Authentication, 2006).

5.6.2 The local database as a user authentication method

Like Telnet user authentication method, this authentication method takes place locally within the NAS itself, and consists of a global configuration of a username along with

a password. The local authentication method is considered stronger than Telnet authentication method, as it depends on a pair of username/password, which represents a stronger approach than the password approach provided by Telnet authentication; and thus somehow protects the user access into the router. Also this authentication methods' username/password are per user credentials stored on the local database of the NAS router, as opposed to Telnet's authentication all-user password that will be so easy to remember and to propagate among hackers once known or exposed to an untrusted user.

Let's consider the code configuration that implements the local authentication method on the NAS router:

The global configuration is followed by a line configuration including a **"login local"** command, which makes the local password override the password configured on the VTY line, and thus takes a remote user connecting through a Telnet connection to verify his credentials against the credentials stored in the local router database, before his access is permitted through the autocommand to reach the organization's inner resources.

```
nas(config)# username user's_name password user's_password

nas(config)# line vty 0 4
nas(config-line)# login local
```

Thus this method is about authenticating a number of users for the Router access. However when we consider the possibility of having hundreds of routers, it would be difficult to maintain all of the user accounts on all of the routers, which poses a scalability problem related to this type of authentication. In order to overcome this problem, AAA authentication servers can be implemented to authenticate the user Exec access (Telnet connection).

5.6.3 The AAA authentication servers as a user authentication method

The AAA paradigm has many advantages and disadvantages when it comes to the remote user authentication process. In this section, we highlight some of the reflections about that type of authentication, while comparing it to the local authentication method (since line authentication method is never recommended to be used within dynamic ACL's connections).

- The AAA paradigm offers great benefits, when it comes to scalability, as it allows the centralization of the user accounts on one (or more) dedicated server(s) containing all the security policies that define the list of users and what they are allowed to do. Thus the AAA server maintains all remote users credentials instead of managing and storing them directly on one or more NAS router. Once users need to access to organization's network, The NAS router(s) sends the authentication and authorization requests (as well as the accounting requests in some cases) to the AAA server to validate them, according to the security policies contained in the user profiles. The NAS router then applies the AAA server validation result permitting or denying users' access to the network.

  Having an AAA centralized user login management is more scalable than configuring and changing the user credentials on many individual switches and routers. Thus the AAA server authentication method is considered as having a better manageability aspect, as well as scalability features, when compared to the local database authentication method.

- AAA server user authentication method depends on using the IEEE 802.1X user authentication standard[55] as a part of the overall Network Administration Control (NAC) strategy, where the user will not be granted access to the organization's inner network, unless he supplies some login credentials to the NAS. Then the NAS sends the user credentials to the AAA server. There will be a series of authentication

[55] 802.1X is an IEEE standard authentication protocol for network access control (Wikipedia - IEEE 802.1X, 2012). 802.1X encapsulates the authentication messages between the user (supplicant) and the authentication server, passing through the NAS (authenticator).

messages going back and forth between the user and the NAS, as well as between the NAS and the AAA server. The AAA server discards all frames on the inner network, except for those 802.1x messages to and from the user. Once the NAS receives messages from the AAA server indicating that the user has been successfully authenticated, all traffic will be permitted between the user and the ultimate destination the user originally needed access to, according to the access permitted by the dynamic ACL entry. With a user unsuccessful authentication message received from the AAA the server, the user will be denied access to the destination stated within the dynamic ACL entry.

- The message exchange between the remote user and the AAA server may sometimes delay the authentication process, especially in case of the existence of some connection problems or network congestions. Thus local authentication, where the NAS checks and decides for the authentication of the user login information, by itself, might be considered a faster way, in this sense, since no authentication messages have to be exchanged over the LAN.
  The message exchange delay possibility, in the case of AAA server authentication used during a Dynamic ACL connection, might cause the user VTY login timeout duration, which is the idle timeout, to expire before the authentication results are sent from the AAA server to the NAS router. This will cause the authentication to close and thus the user has to reconnect through VTY in order to re-authenticate. Which might cause a delay for the whole access process for the user. This can be considered as one reason to favor local authentication method than the AAA server authentication method, as well as it demonstrates that network security always has a conflict with network connectivity.

- AAA server method applies encryption to the exchanged login information between the NAS and the AAA server, an aspect that indicates reliability. In the case of the local authentication method, the encryption aspect need depends on the location of the database: since the database is located on the NAS router itself, then there is no need to encrypt the data since there is no data exchange. This means that the reliability aspect

is still present in the case of the local authentication, even though there is no exchanged data encryption done or needed.

- When it comes to ease of configuration and device manageability, choosing one authentication method, while excluding another, is relative to the size of the network to be protected. Local authentication method should be preferred to the AAA authentication method only when the network size is small, having only one or two NAS router(s) to be configured, updated with users credential information, and maintained for consistency. Since AAA security services involve a certain number of security devices to be configured, and to be contacting each other through certain settings, such a configuration will be more complicated than helpful when compared to configuring/maintaining only one or two NAS routers.

  However the AAA authentication method will be very helpful in case of medium to large size networks, as it will facilitate the management of user credentials, by maintaining them in one or more centrally controlled AAA servers, which will provide ease of manageability as well as consistency between a large number of NAS routers (Beg12; Beg12; Beg12).

- AAA security services provide redundancy amongst the used AAA servers, which allows for risk management in case of failure or inability to reach any of the servers. AAA servers allows the grouping of a number of servers from the same type, so that referencing this group would try the authentication process using one server after another within that group, enforcing redundancy. That redundancy feature is not provided by the local authentication method.

- AAA security services provide a great degree of reliability, for all three services: authentication, authorization and accounting, as it gives the possibility of combining more than one method of each service, in order to allow recovery for the NAS-AAA server connection errors whenever they occur. For instance, when it comes to the authentication process, AAA provides the possibility of defining authentication

method lists, which combine many types of authentication methods for the NAS router to use, in order to secure remote user access to the organization's network, as explained in chapter 4. This feature provides another layer of redundancy, as well; since allowing a number of authentication methods will cover for each other's unavailability. The login access' users, for example, will be authenticated using more than one type of authentication method depending on the configuration and the network engineer preferences. We specified the login access in this study as Dynamic ACLs mechanism mainly depends on it, where the users have to use VTY lines to initiate their access into the NAS router. That reliability feature is not provided by the local authentication method.

As we explained the drawbacks of Telnet (login authentication method) in this chapter, we will show how it can combined with, or substituted by other authentication methods, such as AAA authentication server, using authentication method lists in order to authenticate users' login access (Telnet as an access method). Thus we will introduce some examples of partial configurations to illustrate the use of the different authentication combinations within dynamic ACLs, followed by a differentiation between the different types of AAA server authentications that can be applied within the configuration, comparing them according to many perspectives, including scalability, reliability, ease of configuration, manageability, and cost efficiency.

Example of Telnet implementation as an access method and as an authentication method

In the following example, we show the implementation of AAA authentication along with Telnet as an access method, as well as an authentication method. The detailed configuration and explanation of this example will help the reader identify these two different uses of Telnet, as well as the use of authentication method lists as described in chapter 4.

Example 1 – Line password and TACACS+ authentication

In this example, we configure a dynamic ACL with TACACS+ authentication server protection. The TACACS+ authentication method is configured in a default authentication

method list, which will be automatically applied to the VTY lines, overriding the line authentication (telnet authentication).

Figure 5-3 will help the reader better visualize the topology:

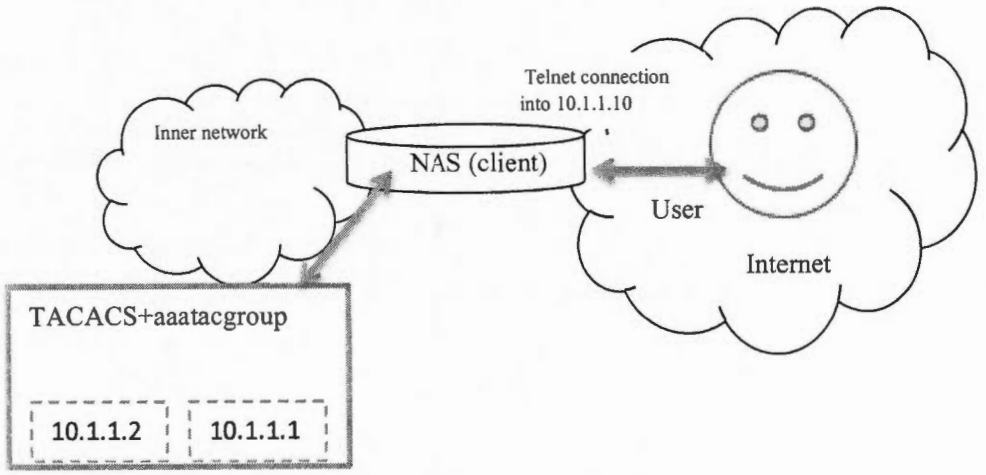

Figure 5.3 The Authentication server (AS)'s peer recovery

The scenario is as follows:

The user attempts to access the NAS interface 10.0.0.10 through a Telnet connection in order to reach the organization's resources. The Telnet connection will be allowed by an extended ACL configured on that router interface, however this Telnet connection will lead the user to an access authentication method. Once the user is successfully authenticated, the Telnet connection drops after triggering an autocommand. The autocommand will open a temporary entry for the authenticated user connection, through the creation of a Dynamic ACL's entry within the extended ACL. The Dynamic entry will allow user's access to the network inner resources.

Important implementation points to be considered:

- The authentication method included within Telnet (VTY as an access method), should be specified right after the reserved word "login", however, in our

configuration, the word "login" doesn't specify any specific authentication method. Since the code includes a method list named "default", that default method will be the authentication method applied to all users accessing the router through all access methods whose authentication methods are not specified, including Telnet access method. Also, this default method will be automatically applied to all router interfaces that are assigned these specific access methods, including Telnet.

Thus Telnet access method will neglect the Telnet password specified within the Telnet (VTY) access method portion and will lead the user to the default authentication method.

- The default authentication method list has only one authentication method specified, namely the TACACS+ method. This method uses the default name "TACACS+", which refers to the whole set of TACACS+ servers that are configured on the NAS router. Since there are two TACACS+ servers configured on the router, then the authentication will be tried on the first server at IP address 10.1.1.1, and if the server is not available to perform that function, the next server at IP address 10.1.1.2 will be tried.

The configuration code is as follows:

```
NAS(config)# aaa new-model                                          (1)

NAS(config)# aaa authentication login default group tacacs+         (2)

NAS(config)# tacacs-server host 10.1.1.1 single-connection          (3)
key secret1
NAS(config)# tacacs-server host 10.1.1.2 single-connection
key secret2

NAS(config)# access-list 101 permit tcp any host 10.1.1.10 eq       (4)
telnet
NAS(config)# access-list 101 dynamic testaccess timeout 15
permit ip any any
```

```
NAS(config)# line vty 0 15                                          (5)

NAS(config-line)# autocommand access-enable timeout 10              (6)
NAS(config-line)# password cisco4dacl                               (7)
NAS(config-line)# login                                             (8)

NAS(config)# interface s0                                           (9)

NAS(config-if)# ip address 172.18.23.9 255.255.255.0                (10)
NAS(config-if)# ip access-group 101 in                              (11)
```

The configuration numbers refer to the following:

1) This command enables AAA globally on all NAS interfaces.
2) This command builds an authentication method list, default, to be applied on the VTY line or any other access method as long as no other authentication method list is applied to it. The access authentication method list contains one authentication method:

   The TACACS+ authentication method (group tacacs+) tries all two TACACS+ servers configured on the NAS router to perform the authentication. Thus the NAS will first try 10.1.1.1, if an error is returned, The NAS will try 10.1.1.2, if an error is returned, the NAS will try the next authentication method on the list, since no more TACACS+ servers are configured on the NAS router. And since there are no more authentication methods configured on the default list, then the user will be denied access to the NAS in this case.
3) These two commands are to identify the TACACS+ daemons to be used with this NAS router: 10.1.1.1 and 10.1.1.2, as well as their shared encryption keys to be used when they communicate with the NAS, respectively: secret1 and secret2.
4) These two commands are to identify the lock and key ACL allowing only Telnet access to the NAS router 10.1.1.10; and the dynamic entry, testaccess, that would permit all access to the organization inner network once the autocommand, configured within the Telnet configuration in part # 5, is triggered by the user

authentication process.

5) This command is to switch the configuration mode from global configuration to vty line configuration; it also identifies the specific lines being configured.
6) This command is to specify the autocommand that will place the dynamic ACL entry into the NAS router configuration, as shown in part # 4. The autocommand will allow user access by creating that dynamic entry only after the user authentication process is successfully performed.
7) This command defines the line password.
8) This command is to make the NAS prompt the user for a login password.
9) This command is to switch the configuration mode from global configuration to interface configuration mode; it also identifies the specific interface being configured, s0. That interface will be the access interface through with the user will access the NAS router.
10) This interface subcommand is to specify the IP address of s0 interface as well as its subnet mask.
11) This command is to apply the lock-and-key ACL 101 configured in part # 4, to s0 interface, to filter all incoming packets on that interface.

As the configuration shows, the default authentication method list contained only TACACS+ authentication method. Actually this method list could have been applied for any access method, other than the VTY (Telnet/line) access included within the configuration, as well as it could have been included other authentication methods besides TACACS+.

According to the configuration, when the user attempts to Telnet access the NAS router, he will be prompted to enter a password, because of command # (8). Though the VTY lines (Telnet) have a password configured, according to command # (7), the TACACS+ authentication, configured as a default login authentication method in command # (2), will override the line (Telnet) password authentication (Cisco IOS Security Configuration Guide, Release 12.2-Configuring Authentication, 2006).

Thus, in this case, the Telnet connected user will be prompted to enter his own credentials as configured within his user profile located on the TACACS+ server(s), in order to

access the NAS router. The Telnet authentication will be used, then, only as a connection method, not as an authentication method, since the authentication will be provided by TACACS+ server(s).

Provided the configuration was missing command # (2), which included the TACACS+ server(s), the Telnet connected user would have been prompted to enter his Telnet password in order to access the router, and Telnet would have been used as both a connection method, as well as an authentication method.

This example was introduced in order to differentiate between Telnet as a connection method for Dynamic ACLs' remote users, versus its use as an authentication method, within the same context.

Now that we explained the authentication mechanism that takes place while implementing Dynamic ACLs, let's introduce some of the Dynamic ACLs security aspects that might represent main concerns when deciding whether to implement Dynamic ACLs as a security solution to protect an organization network.

5.7 Dynamic ACLs security aspects and the scalability issue

Dynamic ACLs provides a great level of security robustness, flexibility to use a variety of authentication methods, reliability of packet transport, interoperability as well as network resource manageability. However, out of all the parameters that have to be considered when evaluating the security performance of Dynamic ACLs as a security technology, scalability is not well supported by dynamic ACLs, according to TacACK (TacACK.com - TACACS+, 2009).

In order to better explain this point, let's consider the following configuration code:

```
access-list 102 permit tcp any host 172.18.21.2 eq telnet          (1)
access-list 102 dynamic testlist timeout 15 permit ip any any      (2)
!
!
```

```
line VTY 0 15                                    (3)
 autocommand access-enable timeout 5             (4)
 password cisco                                  (5)
```

We can notice that the autocommand at command line # 4, includes an "access-enable" subcommand. Actually the "access-enable" subcommand opens up all dynamic ACL on a certain number of connections to the inner network resource. That number is only up to the VTY line numbers permitted by the router, thus only 16 connections (from 0 to 15). Using a scalable AAA server (like TACACS+) doesn't expand that number because the Dynamic ACLs nature doesn't permit it.

Thus Dynamic ACLs' lack of scalability aspect cannot be directly compensated by the scalability aspect of AAA servers (like TACACS+).

5.7.1 The Auth-proxy as a solution for the Dynamic ACLs' scalability issues

In order to improve scalability, the dynamic ACLs concept can be substituted by the use of Authentication proxy (auth-proxy) in a design where per-user-ACLs can be configured on the NAS in order to create the scalability aspect.

The auth-proxy server permits the network engineer to apply security policies through personalized per-user-ACLs. It will act as a virtual server, where every per-user connection (to the inner network resources) can be triggered independently, once the user authentication has taken place. Also, every user is identified and authorized access according to his individual user profile.

The connection to the organization's network starts by an HTTP session where the auth-proxy will check if the user is authenticated. If the user is already authenticated, then no intervention from the auth-proxy will take place. Otherwise, the auth-server will act as a screen (that the user interacts with) between the user and the organization network to authenticate the user, according to Cisco (Cisco - Implementing Authentication Proxy, 2006).

In the following, we will introduce the auth-proxy mechanism, followed by a comparison between the auth-proxy server and Dynamic ACLs followed by the NAS configuration steps applying the use of auth-proxy server in order to reach a per-user-ACLs configuration.

5.7.1.1 Authentication proxy (Auth-proxy) as a substitute for Dynamic ACLs

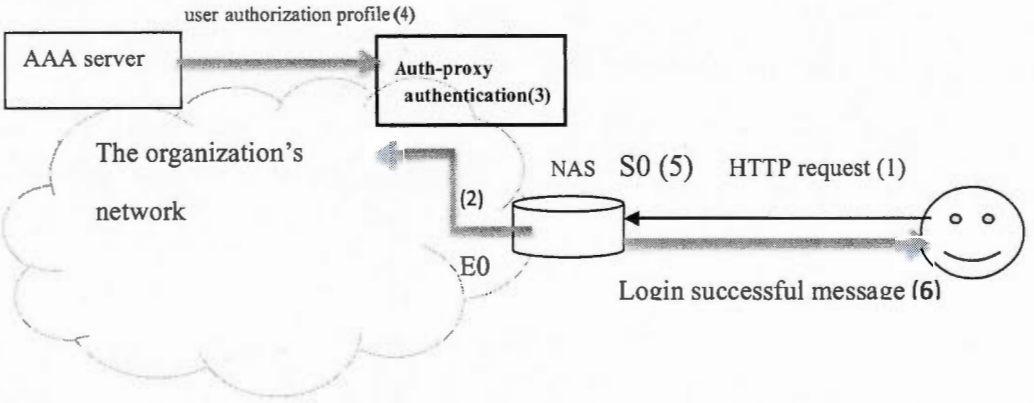

Figure 5.4 Auth-proxy and Dynamic ACLs

According to figure 5-4, the Auth-proxy's mechanism involves the followings steps:

1. The user starts an HTTP session with the organization's HTTP server, through the NAS router.
2. The NAS triggers an auth-proxy server, sending the HTTP request to the auth-proxy server.
3. The auth-proxy checks if the user is already authenticated (by the auth-proxy itself or by another authentication method):
    a. If yes, there exist a valid user authentication, the connection to the inner resource is allowed with no intervention of the auth-proxy.
    b. If no, the auth-proxy prompts the user to enter his credentials, so that the authentication can take place.

4. Once the user is successfully authenticated by the auth-proxy, the user authorization profile is sent from the AAA server (like TACACS+) to the auth-proxy server to check if the access request is eligible for that user.
5. The dynamic ACL entry is built allowing the user access to the inner resource and is added to the NAS access interface configuration. The auth-server actually substitutes the source address, in one or more temporary dynamic ACL entry, by the authorized user IP address.
6. The HTTP server sends a "login successful" message to the user.

When we compare Dynamic ACLs' mechanism to the auth-proxy mechanism, we obtain the following:

1- Dynamic ACLs are triggered through a VTY (Telnet) connection request from the user, while the auth-proxy are triggered through an HTTP connection request from the user.
2- Dynamic ACLs use local authentication, VTY authentication or AAA server authentication; while the auth-proxy uses only AAA server authentication.
3- Dynamic ACLs have both an absolute timeout, and an idle timeout; while the auth-proxy has only an absolute timeout.
4- Dynamic ACLs entries are configured on the NAS router, while the auth-proxy's dynamic ACL's entries are created from the user profile retrieved from the AAA server.
5- Dynamic ACLs access privileges are granted on a per host basis (host IP address), while the auth-proxy's access privileges are granted on a per host basis (host IP address) as well as per-user profile information.
6- Dynamic ACLs entry is only limited to one entry, while the auth-proxy's dynamic entries are multiple, as defined by the user profile information.
7- Dynamic ACLs entry only allows connection coming from a certain host's IP address (or group of hosts). In case there are many users, they all have to connect using that

host (or group of hosts). While the auth-proxy's allows coming connection based on the DHCP-based IP address, which means the free choice for the user to login from any location he needs, independently from the host.
(Cisco - Authentication Proxy, 2008)

5.7.1.2 Auth proxy configuration on the NAS router

Let's consider an example where we need to implement auth-proxy onto the NAS router, the router's serial interface s0 has IP address 10.1.1.10, while its Ethernet interface e0 has IP address 10.10.10.10, and the TACACS+ server has IP address 10.1.1.1.

According to Cisco (Cisco IOS Security Configuration Guide, Release 12.1-Configuring Authentiation Proxy), configuring the NAS router to use auth-proxy involves the following configuration steps:

1) Enable AAA.
2) Define the list of login authentication methods.
3) Define the list of authorization methods.
4) Enable authorization proxy for AAA.
5) Define the HTTP server and enable auth-proxy to use it.
6) Define the security server (TACACS+ in this example) and a server key (TACACS+ or RADIUS).
7) Secure the NAS external interface, by applying an ACL only allowing access to the HTTP server.
8) Enable the NAS internal interface, to receive traffic from the TACACS+ server, by applying an ACL only allowing access to traffic coming from the TACACS+ server.

A configuration code would look like the following:

```
NAS(config)# aaa new-model                                          (1)

NAS(config)# aaa authentication login methlist group tacacs         (2)

NAS(config)# aaa authorization exec group tacacs                    (3)

NAS(config)# aaa authorization auth-proxy default group tacacs      (4)
NAS(config)# aaa accounting auth-proxy default start-stop           (5)
      group tacacs
NAS(config)# ip http server                                         (6)
NAS(config)# ip http authentication aaa                             (7)
NAS(config)#  ip auth-proxy name check-out http list check-http     (8)

NAS(config)# ip access-list extended check-http                     (9)
NAS(config-ext-nacl)#permit tcp any host 10.1.1.10 eq www

NAS(config-ext-nacl)#deny ip any any
NAS(config-ext-nacl)#exit

NAS(config)# tacacs-server host 10.1.1.1 single-connection          (10)
             key secret1

NAS(config)# access-list 102 permit tcp 10.1.1.1 eq tacacs          (11)
      host 10.10.10.10


NAS(config)# interface s0                                           (12)

NAS(config-if)# ip address 172.18.23.9 255.255.255.0               (13)
NAS(config-if)# ip access-group check-http in                       (14)
NAS(config-if)# ip auth-proxy check-out                             (15)

NAS(config)# interface e0                                           (16)
```

```
NAS(config-if)# ip address 10.10.10.10 255.255.255.0          (17)
NAS(config-if)# ip access-group 102 in                         (18)
NAS(config-if)# exit                                           (19)
```

The configuration line numbers refer to the following explanation:

1. This command enables AAA globally on all NAS interfaces.
2. This command builds an authentication method list, methlist, to be applied on vty line or any other login access method. It contains only one authentication method: TACACS+ (group tacacs+), where the TACACS+ servers configured on the NAS router (in part# 5) to perform the authentication. Thus the NAS will first try 10.1.1.1, if an error is returned, then NAS will try the next authentication method on the list, since no more TACACS+ servers are configured on the NAS router, then the user will be denied access to the NAS in this case.
3. This command builds an authorization method list, which allows exec access to the NAS router only if the user is authenticated and only if he belongs to the list of authorized users.
4. This command is to specify the use of auth-proxy authorization to set the user Exec mode, through the use of TACACS+ AAA server. If the TACACS+ server is unavailable, the user is denied access.
5. This command is to create a method list to define AAA accounting on a per-line or per-interface basis. In this example, TACACS+ is the AAA server that will monitor the accounting information, and the router will report these attributes as accounting records to be stored in accounting logs on the security server.
6. This command enables the specification of a particular authentication method for the users reaching the HTTP server through the router. The auth-proxy uses the HTTP server to communicate with the client/host to authenticate the user.
7. This command sets HTTP server authentication method to AAA.
8. This command defines the auth-proxy policy as it allows authentication only to the remote HTTP connections allowed by the ACL named "check-http" mentioned in part #

9.

9. This command is to create the ACL named "check-http", which is used only to allow HTTP traffic to access the NAS external interface S0, in order to be authenticated.
10. This command is to identify the TACACS+ TACACS+ server to be used with this NAS router: with IP address 10.1.1.1, as well as its shared encryption keys to be used when they communicate with the NAS, respectively: secret1.
11. This command is to return traffic from the TACACS+ server to the NAS router. The source address is the AAA server's and the destination 10.10.10.10 is the IP address of the NAS E0 nearest to the AAA server.
12. This command is to switch the configuration mode from global configuration to interface configuration mode, as well as it identifies the specific interface being configured, s0. That interface will be the access interface through with the user will access the NAS router.
13. This interface subcommand is to specify the IP address of s0 interface as well as its subnet mask.
14. This command is to apply the lock-and-key ACL check-http configured in part # 9, inbound to s0 interface, to filter all incoming packets on that interface.
15. This command is to activate the auth-proxy check-out on the external interface s0.
16. This command is to switch the configuration mode from global configuration to interface configuration mode, as well as it identifies the specific interface being configured, e0, which is the nearest NAS interface to TACACS+ server. That interface will be the access interface to TACACS+ information retuned to the NAS router.
17. This interface subcommand is to specify the ip address of e0 interface as well as its subnet mask.
18. This command is to apply the lock-and-key ACL 102 configured in part # 6, to e0 interface, to return information from TACACS+ AAA server into the NAS router.
19. This command is to exit the interface sub-command mode, getting back into the router global configuration.

The authentication server configuration demonstrated, will help highlighting the detailed implementation of auth-proxy as an alternative solution that can substitute Dynamic ACLs, and

will represent an introduction to compare the implementation criterion of such a solution to Dynamic ACLs, as will follow in the chapter.

5.7.2 The rotary command as another solution for the Dynamic ACLs' scalability issues

In this section we address another aspect related to the Dynamic ACLs' scalability and the availability of the VTY lines to be used for other purposes like the NAS router administration.

Dynamic ACLs use the VTY lines to which the configuration is applied; however the number of these lines on a router is only 16 (0 to 15), and using scalable AAA servers, like TACACS+, doesn't expand that number.

However, VTY lines are only used for lock-and-Key user authentication, and once the authentication takes place, the VTY lines are no longer needed for the user connection (through lock-and-key) to continue, since the user no longer needs to be connected to the NAS router, as the dynamic ACL entry will connect him directly to the resource configured within that entry. Thus the VTY line connection is dropped once the user is authenticated.

However, according to (E-Tutorials Lock-and-Key Configuration, 2012), the use of VTY lines for the user authentication part of lock-and-key ACLs keeps these lines busy during the authentication process, which prevents remote access to the NAS router to perform administration tasks. In order to allow remote administration access to the NAS router, a semi-backdoor might be set up into the NAS router as a solution for such a problem. This will be done by creating one additional VTY line to be used for the remote administration. The number of that line will follow the VTY line numbers. Thus, since the ordinary VTY numbers are using 0 to 15 (VTY 0 15), the additional number will be 16 (VTY 16), which liberates the 0 to 15 VTY range for the regular VTY access, including Dynamic ACL authentication, on one hand; and adds to the scalability of the lock-and-key ACLs, by allowing administrative access on the new created line, on the other hand.

This solution can be done through the use of "rotary" command. The command will also remove the Telnet connection for the new created line from port # 23, and will place it onto port # "3000 + the rotary number". Thus if we are adding only one new VTY line, the rotary

number will be 1, and the port number on which the remote administrator will connect will be 3001 (instead of port # 23 that is specified for regular Telnet connections). This solution liberates the Telnet port and adds to Telnet scalability aspect according to (E-Tutorials, Lock-and-Key Configuration, 2012).

A configuration syntax example for the rotary command, would be like the following:

```
Router(config)# line vty last_line_#_plus_1
Router(config-line)# login tacacs+|local
Router(config-line)# rotary rotary_#
```

In this configuration, the authentication method used is TACACS+. An example would be like the following:

```
Router(config)# line vty 16
Router(config-line)# login tacacs+
Router(config-line)# rotary 1

NAS(config)# interface s0

NAS(config-if)# ip address 172.18.23.9 255.255.255.0

NAS(config-if)# ip access-group 101 in port 3000 rotary 1
```

According to this configuration, when the remote administrator needs to access the router, he would Telnet on port # 3001. Thus the router should allow access to this port number in the interface inbound ACL, as shown by the example. In practical world, the use of Telnet is not recommended with both lock-and-key access, and remote administration access; because in order to secure Telnet administration access, it is recommended to allow Telnet to only inner network interfaces, excluding outer network NAS interfaces, which cannot be applied while

lock-and-key access takes place[56].

Now that we explained the security aspects of Dynamic ACLs, emphasizing its scalability issues and ways to overcome them, let's introduce the use of Dynamic ACLs altogether with the AAA servers in order to perform the user authentication process.

5.8 Dynamic ACLs combining different authentication method for recovery purposes

Example 2 – RADIUS TACACS+ authentication combination

In this example, we will show how to configure VTY lines with a method list including three different authentication methods, how the Exec mode authorization process takes place depending the type of the authentication method and how the privilege mode authorization process takes place independently from the type of the authentication method applied by the NAS. The example will be followed by some reflections about the AAA authentication methods used, comparing them in many aspects, in order to pinpoint their strength and weakness.

Thus, in this example, we configure a dynamic ACL with a method list, « methlist », including local database authentication, RADIUS authentication server, and TACACS+ authentication server. The different authentication methods will be used in a peer recovery scenario, AAA security policy is used; and the authentication method list will be applied to the VTY lines, to secure remote user access through them.

The NAS will then authorize the user to reach the user Exec mode, according to the authentication method through which the user was authenticated: thus, for example, if the user enters a username/password at the user prompt, and gets authenticated, the Exec mode authorization method that the NAS will use is the local authorization method, not the RADIUS

[56] Creating additional lines for remote administrators can be modified to allow that access through an SSH connection, which is more secure than Telnet. In this case, the port number to be used will be "2000 + the rotary number".

or TACACS+, as none of these servers would have stored the user authentication data. Therefore the authentication server will be the one responsible for performing the authorization as well.

Then if the user attempts to issue privileged commands, the NAS will attempt to authorize the user to reach the privileged mode, using a different AAA server, Kerberos. This authorization attempt will only take place if the user is successfully authenticated.

```
NAS(config)# aaa new-model                                          (1)
NAS(config)# aaa authentication login methlist local group radius   (2)
                group tacacs+

NAS(config)# aaa authorization exec default local group radius      (3)
          group tacacs+
NAS(config)# aaa authorization command 2 default krb5               (4)
if-authenticated

NAS(config)# radius-server host 172.16.71.146 auth-port 1645        (5)
                acct-port 1646


NAS(config)# radius-server attribute 44 include-in-access-req       (6)
NAS(config)# radius-server attribute 8 include-in-access-req

NAS(config)# tacacs-server host 10.1.1.1 single-connection          (7)
          key secret1
NAS(config)# tacacs-server host 10.1.1.2 single-connection
          key secret2

NAS(config)# username user1 password cisco4dacl                     (8)


NAS(config)# access-list 101 permit tcp any host 10.1.1.10          (9)
eq telnet
```

```
NAS(config)# access-list 101 dynamic testaccess timeout 15
permit ip any any
```

```
NAS(config)# line vty 0 15                                    (10)
```

```
NAS(config-line)# autocommand access-enable timeout 10        (11)
NAS(config-line)# login methlist                              (12)
```

```
NAS(config)# interface s0                                     (13)
```

```
NAS(config-if)# ip address 172.18.23.9 255.255.255.0          (14)
NAS(config-if)# ip access-group 101 in                        (15)
```

The configuration numbers refer to the following:

1- This command enables AAA globally on all NAS interfaces.

2- This command builds an authentication method list, methlist, to be applied on vty line or any other access method, and it contains three authentication methods:

- The first authentication method (**local**) uses the local database username/secret password logins, specified within the **username** commands that are configured globally on the NAS (to involve all NAS interfaces) in part # 8. Since the username local logins are configured on the NAS, then the local authentication method can take place. If there were no local logins configured on the NAS, the local authentication method would have returned an error, and the NAS would have attempted to use the next authentication method on the list to secure the remote user access.
- The second authentication method: RADIUS authentication method (group radius) tries the only RADIUS servers configured on the NAS router (in part# 6) to perform the authentication. Thus the NAS will first try 172.16.71.146, if an error is returned, then NAS will try the next authentication method on the list, since no more RADIUS servers are configured on the NAS router.

- The third authentication method: TACACS+ authentication method (group tacacs+) tries both TACACS+ servers configured on the NAS router (in part# 7) to perform the authentication. Thus the NAS will first try 10.1.1.1, if an error is returned, then the NAS will try 10.1.1.2, if an error is returned, then NAS will try the next authentication method on the list, since no more TACACS+ servers are configured on the NAS router. And since there are no more authentication methods configured on the default list, then the user will be denied access to the NAS in this case.

3- This command is to specify the use of local authentication information to be used to set the user Exec mode if the user authenticates using the username through the local database, it also specifies that the RADIUS authentication information will be used to set the user Exec mode in case the user authenticates using RADIUS, as well as the use of TACACS+ authentication information to set the user Exec mode in case the user authenticates using TACACS+ server.

4- This command specifies the use of Kerberos server to authorize the users reaching the privilege mode whenever they issue privilege mode level 2 commands, only after these users authentication successfully takes place.

5- This command is to identify the IP address of the RADIUS server to be used, as well as its UDP destination port for authentication requests received from the NAS router, UDP destination port for accounting requests received from the NAS router.

6- These two commands are to send RADIUS attribute 44 (acct-Session-ID) and to send RADIUS attribute 8 (Framed-IP-Address) respectively, in access-request packets.

7- These two commands are to identify the TACACS+ daemons to be used with this NAS router: 10.1.1.1 and 10.1.1.2, as well as their shared encryption keys to be used when they communicate with the NAS, respectively: secret1 and secret2.

8- This command specifies the username and password to be used with the local authentication method.

9- These two commands are to identify the lock and key ACL allowing only Telnet access to the NAS router 10.1.1.10; and the dynamic entry, testaccess, that would permit all access to the organization inner network once the autocommand, configured within the Telnet configuration in part # 10, is triggered by the user authentication process.

10- This command is to switch the configuration mode from global configuration to vty line

configuration, as well as it identifies the specific lines being configured.

11- This command is to specify the autocommand that will place the dynamic ACL entry into the NAS router configuration, as shown in part # 9. The autocommand will allow user access by creating that dynamic entry only after the user authentication process is successfully performed.

12- This command is to apply the authentication method "methlist" to the VTY lines, specifying how authentication should be done to secure user access through the VTY line.

13- This command is to switch the configuration mode from global configuration to interface configuration mode, as well as it identifies the specific interface being configured, s0. That interface will be the access interface through with the user will access the NAS router.

14- This interface subcommand is to specify the ip address of s0 interface as well as its subnet mask.

15- This command is to apply the lock-and-key ACL 101 configured in part # 4, to s0 interface, to filter all incoming packets on that interface.

In this example, we can see that the AAA paradigm is applied through command # (1) in order to differentiate, configuration and implementation wise, between the three AAA services. The peer recovery will take place between the Local authentication, RADIUS and TACACS+; while the user attempts to authenticate reaching the user Exec access mode, as shown in command # (2). The authorization to reach that mode will depend on the authentication method used. E.g., if the first authentication method is available, the user will be prompted to enter a local username/password in order to be authenticated through the local database. If the user is successfully authenticated, the user will be authorized locally also, and according to the rules specified locally on the NAS router. However if the local authentication method is not available, the peer recovery mechanism will lead the authentication process to follow the next method stated in the method list in command # (2), namely RADIUS, and the user will be prompted to enter his credentials as specified upon the RADIUS server. In this case, if the user becomes successfully authenticated by RADIUS server, the local authorization method

(specified in command # (3)) will fail, since no data is saved from the local authentication. The authentication will then be done through RADIUS server according the user profile policies associated with that user. The scenario will be applied following the same logic with the TACACS+ server, in case the other methods were not available. Thus we can see that, in order to allow access to the Exec mode, the authentication method will define the authorization method to be used.

When we come to the user-Privileged mode, things will be different, since the user has to be successfully authenticated and authorized to access the Exec mode before he attempts to access the Privileged mode. Thus the authentication process already takes place before the user is allowed access to the user Exec mode, and since the user has already access to the network through the Exec mode, then no more authentications are needed. Therefore, in order to authorize the user to access the Privileged mode, it doesn't matter which authentication method was used to authenticate the user reaching the Exec mode. What really matters is whether the user is authenticated and authorized to access the user exec mode. Thus the authentication process done in the Exec mode is separate, and does not define the authorization process of the Privileged mode.

Thus if the a user is authenticated and authorized to use the Exec mode, and if he attempts to reach the Privileged mode in order to issue administrative commands, then the user will be authorized through the authorization method list "default" specified in command # (4), hence through Kerberos server, regardless of the method that was used for the authentication.

This example has explained the difference between the user Exec mode, and the user Privileged mode, in terms of defining the authentication and the authorization processes to be used in order to allow successful user access to the organization's resources.

5.9 Comparison of AAA servers

Now that we introduced different AAA servers used for authenticating remote user access through the Dynamic ACL's VTY lines, let's compare theses servers in order to emphasize their strengths and weaknesses in terms of efficiency, scalability, reliability, interoperability, ease of configuration, and manageability.

5.9.1 RADIUS versus TACACS+

Let's consider RADIUS, which is an IETF standard, and TACACS+, which is described in RFC 927 and 1492 as an informational standard only. Feature wise, TACACS+ can be considered as an extension for RADIUS as both give the same functionality, however TACACS+ changed RADIUS implementation to meet the needs of modern networks. Some key differences exist between TACACS+ and RADIUS specially the transport reliability, the encryption, the separation of the authentication and the authorization functions, multiprotocol support, interoperability, router management, performance impact and implementation priorities. Let's demonstrate each of these points of difference in the following:

1- In terms of reliability and scalability, TACACS+ uses TCP, while RADIUS uses UDP as a transport protocol that help transport datagrams between the client host and the AAA server, which affects performance and the reliability of the transport layer. TCP offers many advantages over UDP, since it is more reliable protocol, always using acknowledgment (of received requests) and synchronization no matter how loaded the backend authentication mechanism is. Also TCP provides a connection-oriented transport of datagrams, while UDP only provides best-effort delivery, and is always susceptible to situations like network congestion and server crashes. Even with additional programmable variables that can be added to RADIUS, to overcome the UDP delivery drawbacks, RADIUS will always lack the built-in support that the TCP offers. TCP can immediately detect if the server is down, slow or non-existent, and can maintain many simultaneous server connections only addressing the ones that are functioning, while UDP cannot differentiate between server statuses. TCP is more

scalable to network growth and congestion (E-Tutorials RADIUS AND TACACS+ comparison, 2012).

2- In terms of security, TACACS+ can encrypt the whole access request packet (the entire session) coming form the NAS to the AAA server, which happens during normal operation to ensure the packet security, though the packet can stay unencrypted for debugging purposes. RADIUS encrypts only the password, leaving other critical information like the username, authorized services and the accounting information unencrypted and subject to security attacks, according to Slaptijack (Slaptijack, 2012).

3- In terms of flexibility, TACACS+ separates the three AAA functions, while the authentication (to allow the host/device access) and authorization (to allow the user access) functions have to be combined in RADIUS packets sent from the server to the NAS. TACACS+ separation for the three AAA functions can even allow the use of different servers to address each function (e.g. the combination of Kerberos authentication along with TACACS+ authorization and accounting.) This decoupling of the authentication function separately off the authorization and accounting allows the NAS control over the commands that can be executed by the user, as the NAS will frequently ask the TACACS+ server for additional authorization information in order to grant the user the permission to use a particular command on the NAS.

4- In terms of flexibility, while TACACS+ provides multi-protocol support, RADIUS doesn't support AppleTalk Remote Access (ARA), NetBIOS Frame Protocol Control, Novell Asynchronous Service Interface (NASI), or X.25 PAD connection protocol.

5- In terms of interoperability, due to the Cisco proprietary nature of TACACS+, interoperability cannot take place. Though it supports many protocols as previously mentioned, it is supported by a large number of vendors. RADIUS that is supported by all vendors. Though different vendors implement RADIUS, RADIUS doesn't guarantee interoperability as interoperability will only exist when different vendors implement the same attributes, even though in real implementations, many vendors implement extensions that are proprietary attributes which interferes with the possibility of interoperability, however, in practice, RADIUS works well in a mixed vendor environment.

6- In terms of manageability, TACACS+ allows a fine grain level of user commands, permitting more controlled access for a greater number of users on the network. Actually, TACACAS+ authorization of router commands (per-command-authorization) can be done either on a per-user (as it supports 15 privilege modes) or per-group basis, thus it allows router management and flexible terminal services. Also TACACS+ that allows all AAA functions to take place, where both exec and command controls can be implemented. RADIUS has a limited control over the user privilege mode (E-Tutorials RADIUS AND TACACS+ comparison, 2012). Also, RADIUS doesn't allow command authorization or command accounting, though it allows exec authorization and exec accounting.

7- In terms of router manageability and performance, due to the greater number of messages exchanges it uses between the NAS and the server allowing authentication, TACACS+ is heavier than RADIUS on the NAS router performance wise. TACACS+ exchanges both exec and command authorization as well as both exec and command accounting[57]. On the contrary, RADIUS has less performance impact on the router than TACACS+ as it uses less memory and CPU cycles on the NAS router, due to the fewer number of commands and messages exchanges (only exec authorization and exec accounting messages) it uses between the NAS and the server on one hand, and the user on the other hand.

In general, TACACS+ should be used when the priority is security and flexibility, within a Cisco based network. Security is accomplished with TACACAS+ since the full session is encrypted, and since authorization and authentication are separate. The flexibility is through the use of TCP, which allows more router manageability in more advanced networks. Flexibility is through TACACS+ support for more enterprise protocols as well as through the availability of full 15 privilege classes to control router command prevention. RADIUS should

[57] TACACS+ allows both exec and command accounting as well as both exec and command authorization to take place while the user tries to access an organization's inner resource. However RADIUS only allows exec authorization and exec accounting, which is one of the reasons why RADIUS messages communicated between the server and the NAS are much less in number and size than TACACS+ messages communicated between the server and the NAS.

be used when the priority is availability, where RADIUS or a combination of RADIUS and TACACS+ can be used on a mixed vendor network not supporting TACACS+ protocol.

Now that we introduced the difference between TACACS+ and RADIUS, let's compare RADIUS with DIAMETER.

5.9.2 RADIUS versus DIAMETER

DIAMETER is an IETF standard, described in RFC 3588 ad 3589. DIAMETER is considered as an upgrade for RADIUS overcoming some of its limitations, though it is not compatible with it.

As TACACS+ and RADIUS were developed to address dial-up Point-to-Point and terminal server access, DIAMETER was developed to address new access technologies like wireless, DSL, Mobile IP, and Ethernet, as well as the increasing complexity of the NAS routers. Some of the improvements introduced by DIAMETER protocol as compared to RADIUS are as follows.

1. Transport reliability mechanisms through the use of TCP or SCTP, as a transport protocol within the Transport layer of the OSI network architecture model; versus UDP used by RADIUS[58].
2. Transport security mechanisms through the use of mandatory IPSEC and optional Transport Layer Security (TLS) support are used in DIAMETER though TLS is not yet standardized for RADIUS.
3. Application layer acknowledgements and failover algorithms.
4. Server message acknowledgment (server-initiated messages) is supported so that the NAS would detect whether the server has received the request or whether the message was silently discarded as the server sends error messages, authentication and a session termination message. In case of RADIUS, the request status is unknown due to the lack of such messages.

[58] The IETF is in the process of standardizing The TCP transport protocol for RADIUS.

5. Both end-to-end as well as hop-by-hop-security are supported, to guarantee the information remains unmodified throughout the session. RADIUS supports only hop-by-hop security, where every hop can modify information that cannot be traced to its origin.
6. Capability negotiation between the client and the server.
7. Better roaming support between Access points in case wireless technologies are used.
8. Data object security[59] is supported but not mandatory.
9. Easier to extended, adding new attributes and commands to be defined, as it supports both vendor-specific attributes (VSA) as well as commands; while RADIUS supports VSAs only.
10. The length of user data fields which contains the user information (like his SSN number) and are usually used in the accounting logs, is two bytes allowing for a maximum of 16,535, which allows more detailed information about the user; while in RADIUS the length of data field is only 1 byte allowing for a maximum of 255, which implies less detailed user information.
11. Server inaccessibility problems are solved much faster as well as server status failover detection. RADIUS doesn't detect if the server is out of operation, and has to send many requests (3 of them) to the primary server, and if no reply, it has to reroute the request to peers.
12. Dynamic peer discovery through Domain Name Server Service (DNS SRV) and the resource record Name Authority Pointer (NAPTR), as well as peer configuration.
13. Better traffic control within congested networks, thus this is where DIAMETER is recommended, while RADIUS provides fast user Identification with fewer packages, however it is unable to control the traffic (load balancing and dealing with congestions) and its peers (in communication chains) in cases of overcrowded networks.
14. In some implementation cases, session establishment using DIAMETER is 40 percent longer than session establishment using RADIUS.

---

[59] Data Object security uses OSI layer 7 data objects and function codes to determine remote access authorization, increasing the security functions of other solutions like firewalls, encryption and authentication (Mander, Cheung, & Nabhani, 2010).

Now that we have compared RADIUS with both TACACS+ and DIAMETER[60], we can separately specify some of the comparison points related to Kerberos, since Kerberos functions in a slightly different way.

5.9.3 Kerberos

Some of the Kerberos limitations are as follows:

- Kerberos stores all passwords encrypted with a single key. Actually, Kerberos authentication depends entirely on the knowledge of passwords that are encrypted with the server's master key, using conventional algorithms (usually DES). The passwords as well as the master key are stored on the same server, so that the server can decrypt them when needed. This means that in case the Kerberos server or its redundant replicate get attacked, all the stored passwords have to be changed.
- Kerberos doesn't use public key cryptography. Users prove their identity by their knowledge of their password's key. Thus, decrypted passwords can be accessed by the server, which is a disadvantage of Kerberos, since the server's password can be stolen unless the server is both physically and computationally secure. However, public key cryptography can be integrated into Kerberos system through combining public key smart cards, as implemented in Microsoft's Kerberos.
- Kerberos requires a secure server, as Kerberos servers are stateless, having permanent state while keeping user passwords in the Random Access Memory (RAM) rather than on disk, and doesn't need to be updated while the Kerberos transaction takes place. The server has to answer users authentication requests and issue granting tickets, which is a simple design that allows the creation of replicated servers for redundancy; however, each of these servers needs a complete copy of the entire Kerberos database,

[60] Note also that several Diameter IETF drafts are currently in progress.

which also means the necessity of its physical and computational security as well, which in turn, is an administrative and a financial burden.

- Kerberos requires a continuously available server, for the network to be usable. Without a redundant replicating server, if the main Kerberos server is down, no access will be allowed to the organization's resources.
- Every network service/program must be modified in order to be used with Kerberos, an action called "Kerberizing the application", modifying its source code.
- Kerberos doesn't work well in time-sharing environment where many users share a workstation, since Kerberos keeps the tickets in a temporary directory to overcome the difficulties of sharing the ticket between the different processes. And as one user might be granted a ticket, that ticket can easily be stolen by other users sharing the workstation, while they should not be allowed access to the organization's inner resources.
- Kerberos doesn't protect against Trojan horses attacks since many of the workstations contain local copies of the programs that they run, even within a networked environment. Trojan horses are modifications made to the system's software (the workstation) so that the attacker can easily gain access to the system, modifying the user's files, denying access to the authentic user. And since the workstation doesn't authenticate itself to the user, the user could never tell in advance, if the computer became compromised in the first place.
- Kerberos authentication might go beyond its granted time, since the process involves many parameters specified by both the authentication server and the user's workstation. The user is authenticated for the life-time of the ticket coming from Authentication Server, which is the minimum of either: the default time to use an inner service, or the remaining time of a TGT ticket already created for that user. So if the software on the workstation is modified by an attacker before the user attempts to authenticate to access the network, the user authentication might go beyond the ticket life-time, which would allow the user access for a longer time than this user is permitted to (Steiner, Neuman, & Schiller, 1988).

- Kerberos use might lead to a cascading loss of trust, since once a user or server password is disclosed, it is possible for an attacker to use it, decrypting other tickets, and thus gaining more access spoofing inner servers and users.

Based upon the comparison made between the different authentication technologies that can be used within the context of Dynamic ACLs implementation, we introduce some suggestions that emphasize these differences in order to help the reader better choose, based upon certain criteria, one authentication technology suiting his organization's needs.

5.10 Some authentication suggestions based upon the comparison

In this section, we suggest some of the authentication technologies, combined with an access method, that can take place at one point at a time, once the authentication method list configured upon the NAS, gets to try an available authentication method.

Certainly, our suggestions are not to be considered as a universal solution to meet all Lock-and-Key authentication needs of every organization. However our suggested solutions might give a clearer emphasis to the reader, on the flexibility of choosing different lock-and-Key authentication technologies, each having its own pros and cons, and the possibility of combining them with either Telnet or SSH as a connection method, where they can meet a certain organization's security needs.

The authentication methods/technologies suggested are Enable, TACACS+, RADIUS, DIAMETER, and Kerberos. Also, though it is not introduced as a technology to be combined within the Dynamic ACLs authentication process, auth-server is included with our parameters as a suggested technology that can substitute Dynamic ACLs authentication altogether. Since Telnet as an authentication method has a considerable amount of drawbacks, as we explained at the beginning of this chapter, we will exclude it from the list of authentication methods to be compared.

Implementing our suggested solutions will be according to the individual needs of each organization, taking into consideration the design of its own network topology, including all

the criteria that might vary according to such a design. The criteria are security efficiency, flexibility of personalizing authentication process and interacting with other processes, scalability for network growth and avoiding congestions, reliability of transporting the authentication information, interoperability with other vendor devices and protocols, manageability of the network resources including the NAS router and ease of implementation within a given network.

The suggested solutions are represented in a table scaled from 1 to 5 plus signs, where 5 plus signs stand for the highest level of performance demanded for a certain parameter, and one plus sign stands for the least.

The table can be considered as a guideline to weigh the pros and cons of each authentication method regarding each criterion, on one hand, and to weigh the use of auth-proxies as an alternative authentication technology that can replace the use of Dynamic ACLs (regarding each parameter), on the other hand.

The suggested solutions are as shown in table 5-1:

	Enable and Telnet	TACACS+ and Telnet	TACACS + and SSH	RADIUS and Telnet	DIAMETER and Telnet	Kerberos and SSH	Auth-proxy
Security	++	+++	+++++	++	+++	+	+++++
Flexibility	+	+++++	+++++	++++	+++++	++	+++++
Scalability	+	++++	++++	+++	++++	++++	+++++
Reliability	+++++	+++++	+++++	+	+++++	+++++	+++++
Interoperability	+++++	+	+	+++++	+++++	++	+++++
Manageability	+++++	++++	++++	+++	+++++	++++	+++++
Ease of implementation	+++++	+++++	+++++	+++++	+++++	+	+++++

Table 5.1 Suggestions of authentication technologies

As we can see from table 5-1, the indications we can obtain considering the suggested technologies are as follows:

1. The security criterion:
   - The use of Telnet with any of the authentication methods mentioned (including authentication servers and Enable method) degrades the performance of such combination when used to authenticate the Dynamic ACL's remote users, due to the Telnet drawbacks explained in this chapter.

     However the use of SSH increases such a performance level as we can see with TACACS+ when combined with SSH; which will give the same security performance as the one resulting from the use of an auth-proxy by itself.
   - We can notice that the Enable and Telnet, as a combination, is almost as secure as RADIUS and Telnet combined, since Enable, which is the local authentication method performed locally on the NAS router, involves password hashing leaving the username as plain text, which is almost equivalent to RADIUS security nature, encrypting only the passwords while leaving the rest of the users information unencrypted.
   - Also the security performance of RADIUS and Telnet, as a combination, is less than any other authentication server when combined with Telnet; since RADIUS doesn't encrypt the full session when communicating with the NAS within the organizations' inner network and doesn't separate the user authentication and the user authorization processes, which diminishes its security aspects.
   - And the security performance of Kerberos and SSH, as a combination, is less than any other authentication server when combined with SSH; since Kerberos doesn't use public Key encryption for the stored passwords, doesn't protect against some security risks, uses an authentication that might go beyond the granted time needed for the user session, and can sometimes lead to cascading loss of trust problems, which diminish its security aspects.

     This combination's performance, though it uses the more secure SSH as a connection method, is even less secure than using Telnet combined with other authentication servers like DIAMETER or TACACS+, as the SSH's secure connection will not compensate for such security issues related with Kerberos authentication process.

2. The flexibility criterion:
   - We can see from table 5-1 that the flexibility performance of TACACS+ when combined with either Telnet or SSH, as a user access method, is weighing more than the RADIUS and Telnet suggested combination, due to the fact that TACACS+ supports many protocols like NASI, Net BIOS and AppleTalk commands, while RADIUS lacks that aspect. Also the flexibility of TACACS+ lies in the fact that it provides 15 privilege classes which allows a fine-grained control over the user commands to be performed on the NAS router. One more aspect is the use of TCP as the transport protocol in TACACS+ servers, which allows more router manageability in advanced networks. Such flexibility aspects are not provided by RADIUS.
   - Also, the flexibility performance of all AAA server combinations, as well as the auth-proxy's is generally weighing more than for the other combination flexibility performance; since they all take into consideration the user's security privileges included within the user profile, which provides a great degree of flexibility, non existent in the other suggested combinations.
   - We can also see that the flexibility performance of Enable method when combined with either Telnet or SSH, as a user access method, is weighing less than any of the other suggested methods as Telnet doesn't take the user profile into consideration allowing access to each user according to his individual security privilege; thus we cannot consider the local authentication method (Enable) as a flexible suggestion.
   - When we come to Kerberos and SSH combination, we can see that their flexibility performance is less than any other authentication server; since Kerberos requires a source code modification (Kerberizing) for every network service/program, before integrating it within the system. Also because of its security nature, Kerberos doesn't work well in time-sharing environment where many users share a workstation. These two aspects considerably degrade Kerberos flexibility performance. However, we know that Kerberos does take the user profile into consideration, thus its flexibility performance is a little higher than the local authentication (Enable) method.

3. The scalability criterion:
   - We can distinguish that the scalability performance of all AAA server combinations, as well as the auth-proxy's is generally weighing more than the Enable and Telnet combination's, since they all allow a great level of scalability, by nature. However, we can notice that the auth-proxy's performance in this aspect might be slightly higher. This is due to the nature of Dynamic ACLs in general, as they are not scalable, which diminishes the scalability performance of any of the authentication methods, when they are combined with the Enable method, which is not scalable.
   - We can also note that TACACS+, DIAMETER, Kerberos and auth-proxy combinations are more scalable than RADIUS's, since they all use TCP protocol, which provides more scalability to network growth and congestion.
   - It is also worth mentioning that TACACS+ level of scalability is not the maximum that TACACS+ combination can reach, since TACACS+ requires the exchange of a large number of messages (for both exec and command authorization, as well as for both exec and command accounting) between the NAS and the server, which creates a big load on the NAS. The NAS router becomes even more overloaded when the number of the connected users, and the network services to be performed increase, since the number of the exchanged messages increases even more, affecting the NAS manageability and performance.

4. The reliability criterion:
   - We can observe that the suggested combinations that include DIAMETER, Kerberos or TACACS+ weigh higher regarding the reliability aspect, since they use the reliable TCP transport protocol to communicate with the NAS and the other devices involved in the authentication process; versus RADIUS that uses the best effort UDP protocol to perform such communications.
   - Though it replaces the use of Dynamic ACLs and cannot be combined with them, the auth-proxy is considered very reliable alternative, as it also uses TCP transport layer protocol.

- Since the local authentication method (Enable) doesn't communicate within other devices during the user authentication processes and doesn't need to use any transport layer protocols, it is considered as a reliable authentication method.

5. The interoperability criterion:
   - We can see that the suggested combinations that include TACACS+ weigh less than all the other AAA combinations (including the auth-proxy) regarding the interoperability aspect, since TACACS+ authentication method is Cisco proprietary, performing well only within Cisco environments.
   - RADIUS should be used when the priority is availability/interoperability, where RADIUS or a combination of RADIUS and TACACS+ can be used on a mixed vendor network not supporting TACACS+ protocol. Using such combination causes a splitting of references, assessments, designs, etc. Thus, this combination might be considered as a recommended approach when applied with some precautions.
   - Since the local authentication method (Enable) doesn't communicate within other devices during the user authentication processes and doesn't need to be interoperable, it is considered as an interoperable authentication method.
   - We can distinguish that Kerberos's combination is not interoperable as well, since all devices and programs have to be "Kerberized" before their integration within Kerberos authentication process, which is considered as an extra necessary step towards interoperability. This step places Kerberos combination as more interoperable than TACACS's.

6. The manageability criterion:
   - We can see that TACACS+ combinations indicate a great level of manageability, since TACACS+ uses TCP, which can immediately detect if the server is down, slow or non-existent, and can simultaneously maintain connections with many servers. Also TACACAS+ provides full 15 privilege classes to control router command prevention, which allows more router manageability in more advanced networks. Besides, TACACS+ provides support for enterprise protocols like NetBIOS and AppleTalk,

which allows more manageability of the authentication processes throughout the organization's network.

- We can also see that RADIUS combination does not provide a maximum level of manageability. RADIUS is lighter on the NAS router than TACACS+, since RADIUS uses less memory and CPU cycles on the NAS router, due to the fewer number of commands and messages exchanges between the NAS and the server on one hand, and the user on the other hand. However, RADIUS doesn't use TCP neither supports a different range of enterprise protocols; which diminishes its manageability weighing more than TACACS+ combination's.
- Also Kerberos combination, though it uses TCP, it doesn't support a different range of enterprise protocols; which increases its weighing more than RADIUS combination's, regarding that parameter.
- Also DIAMETR combination provides a great level of manageability, as DIAMETER was developed to address new access technologies like wireless, DSL, Mobile IP, and Ethernet, as well as the increasing complexity of the NAS routers. Also DIAMETER uses TCP protocol, which allows more router manageability during the authentication process.
- We can also observe that the Enable combination provides a great level of manageability, as all the user passwords will be included within the router's local database, and there will be no need to communicate with other authentication devices over the network; which eliminates possible issues and allows for even more router manageability.
- In case of the auth-proxy alternative, the manageability provided is at higher level since auth-proxy uses TCP, on one hand, and on the other hand, it supports a variety of network protocols as well.

7. The ease of implementation criterion:
   - We can distinguish that all suggested combinations are very easy to implement and maintain, except for the combination that includes Kerberos, since Kerberos is a very demanding authentication system that relies on its own rules. Thus in order for every device or software to join this system, it will be required to have his source code

modified in order to integrate successfully within. That kerberizing process makes it so hard to implement Kerberos within any environment; as it makes it hard to expand and maintain the hardware or the software of the Kerberos system.

Now that we explained the differences between the types of AAA authentication servers, let's focus on the limitations occurring while using Lock-and-Key ACLs in general, while emphasizing the different ways we can overcome them.

5.11 Dynamic ACLs drawbacks

1. Dynamic ACLs create a security opening in the border firewall/NAS, which makes that router susceptible to Source Address Spoofing. This case happens when the router's interface gets temporarily configured to permit access to the legitimate remote user (source address) entered in the dynamic ACL entry, and thus the legitimate user is allowed access to the organization inner network, and can exchange data through the firewall router. Then a malicious attacker might spoof that legitimate source address, and replace his rogue source address by a legitimate one, in order to gain access to the organization's inner network, even behind that firewall router.

   The source address-spoofing problem is a security problem common to the use of ACLs in general, whether the authentication method used by the ACLs is Telnet or AAA servers. An alternative solution that may help address this drawback is to apply authentication proxies instead of Dynamic ACLs.
2. The Lock-and-key has some security issues, as it specifies the source IP addresses of the hosts not the users, and the security issues can be addressed by the use of Authentication servers.
3. The Lock-and-key has some flexibility issues as it specifies the destination IP address by a maximum of one dynamic entry, while authentication proxies permit many dynamic entries. The flexibility issues can be addressed by the use of Authentication servers.

4. The Lock-and-key has some scalability issues as it can secure only a limited number of user connections, versus multiusers connections (ex: broadband), the scalability issues can be addressed by the use of Authentication servers.
5. The Lock-and-key dynamic entry is not automatically deleted once the user's session is terminated as it remains within the Lock-and-key until the dynamic timeout (either idle or absolute timeout) is reached, until the network engineer manually clears it out, or until the NAS router reboots, whichever happens first[61]. Thus the possibility of having the dynamic entries remaining active on the NAS interface represent a security hole that can be subject for malicious attacks, since an attacker can wait until the legitimate user leaves the machine and use that same authenticated connection to reach the organization inner resources.
6. Since Lock-and-key depends on Telnet in the majority of implementations, it is susceptible to IP spoofing, due to the lack of encryption. As the user authenticates, the Lock-and-key opens up a temporary access hole in the NAS, and if the attacker knows the user's source IP address, he can use it for a spoofing attack. In order to avoid such attacks, encryption might be considered, through the use of SSH or Virtual private networks (VPN).
7. The Lock-and-key connection starts by a Telnet session that serves the goal of user authentication process. This Telnet session keeps port #23 busy, and thus, in case all 16 VTY lines are occupied simultaneously by different user connections, any other access for the router through VTY lines would be prevented, including connections performed for remote router administration. The Rotary variable added to the VTY line configuration can address this problem, as it will represent a semi-backdoor into the router by shifting the lines numbers assigned for Telnet connections, freeing the original VTY lines as well as port 23, for other administrative connections. The rotary solution can also be applied in case of SSH to free NAS port #22, for remote administrative connections (Telnet administrative access is recommended to be

[61] If the NAS router configuration is saved, the Dynamic ACLs entries will not be saved, and will not still exist within the configuration once the router reboots again (unless the user re-authenticates by then).

prohibited on external interfaces, thus only allowed to the network inner devices/users).

8. Even when used with AAA server authentication, Lock-and-key ACLs are not scalable since "access enable" allows access to all authenticated users, and there is no possibility to create per-user ACLs. Lock-and-key scalability issues can be addressed by the use of Auth-servers.
9. The Lock-and-key allows all authenticated users to access certain network resources as specified in the Dynamic entry of the Lock-and-key, since it defines only a limited number of user access policy statement per ACL, thus it is so restrictive in implementing personalized access to such allowed resources, according to each user policy, or according to a group policy, which limits its usefulness in a LAN environment. The use of auth-proxy addresses this issue, setting up per-user-access policies as it allows every per-user connection to be triggered independently, according to his own user profile, allowing the combination of a number of user policies within the same ACL.
10. Another solution would be the use of Lock-and-key to restrict access to a bastion host to which the external users can log in, in order to access inner network resources. Since this solution is less practical and has some limitations, many companies prefer the implementation of the authentication proxy solution.
11. A third solution would be the implementation of double authentication where remote users start their PPP connection authentication through CHAP then the only allowed privilege would be a Telnet connection where the user has to reenter different login information as a second authentication point. Through the Telnet connection, the AAA server would be triggered to allow users privilege according to the user's profile defined within the AAA server used.
12. Lock-and-key impacts the NAS router performance in the following ways:
    - When the dynamic ACL entry is triggered, the Lock-and-key ACL is rebuilt (including that new temporary dynamic entry) on the silicon switching engine

(SSE)[62] of the NAS, which causes the SSE switching path to slow down momentarily.

- The dynamic ACL entry of the Lock-and-key ACL requires the use of the timeout facility, even if the timeout is set to its default value, and consequently, it cannot be SSE switched by the NAS router, and will be processed by the protocol fast-switching path.
- The temporary dynamic ACLs' entries are dynamically added to and removed from the Lock-and-key ACL that is configured on the NAS border interface, which will cause it to grow and shrink dynamically in return. This can degrade the packet switching performance of the NAS router, creating additional performance problems. In some cases, the network engineer should check for these dynamic entries and remove them in order to diminish such problems.

Since auth-proxy seems to be a good solution to address most of the Lock-and-key drawbacks, why do we favor Lock-and-key? Lock-and-key is usually used when authentication proxies are not available or when there is a small number of users to be connected to a bastion server where there is no need for an authentication proxy.

5.12 Recommendations

According to the research performed during this study, we can provide the reader with the recommendations that we have formed based upon the information acquired during our research. The recommendations include some general application guidelines and useful tips to be taken into consideration while configuring Lock-and-key onto an organization's NAS router.

These recommendations will be followed by more specific implementation suggestions, that are given based on the general guidelines included within the recommendations. The recommendation can be summarized in the following points:

[62] SSE is the routing and switching mechanism that compares layer-2 or layer-3 headers of the incoming packets to a silicon switching cache located on the router, and determines the actions required to forward the packets out of the proper router interface (Cisco - Internetworking Terms (SSE), 2011).

- Only one dynamic entry can be specified in the Lock-and-key ACL. Any other dynamic entries will be ignored, or flagged as invalid, depending on the Internetworked Operating System (IOS) version.
- This entry has to be placed towards the beginning of the ACL, so that it doesn't get overridden by other entries preceding it; since this may cause access denial to the users.
- A unique name must be used for the Lock-and-key ACL, specified with the dynamic parameter.
- When specifying the destination within the dynamic ACL entry, it is recommended to use a specific IP address rather than the key word (any), which will indicate the resources to be accessed by the authenticated users.
- One of the Lock-and-key static ACL entries should allow either Telnet or SSH, in order for the users to get authenticated.
- At least one timeout should be configured: either as an absolute timeout specified within the dynamic ACL entry, or as an idle timeout within the auto-command configuration. If neither timeouts is configured, the Dynamic entry will remain forever within the Lock-and-key ACL, representing a security hole, until manually removed by the network engineer.
- If both timeouts are present, the idle timeout must be less that the absolute timeout, otherwise the IOS might face issues while removing the dynamic ACL entry off the configuration.
- The Lock-and-key ACL entries might be combined with timed entries, so that remote users access for authentication would only take place within specific times, which adds more prevention to possible security threats.
- The Lock-and-key ACL can be applied to restrict access coming from external remote users to inner resources, and to restrict access from internal users to outside resources. The configuration of the dynamic entry will remain the same in both cases. However, in the first case, the ACL will be applied inbound for the traffic entering the NAS serial (border) interface, and the word "any" will be replaced by the remote user/device's IP address accessing the destination inner host. And in the second case, the ACL will be applied outbound for the traffic exiting that same NAS serial (border)

interface, and the word "any" will be replaced by the inner user/device's IP address accessing the destination external host to be denied access.

- The Telnet or SSH can be configured using the rotary number in order to avoid the overuse of the VTY lines; which will permit administration access to the NAS router using the VTY lines.
- Both SSH and Telnet are supported as a line connection for the Lock-and-key ACL authentication process. However, since SSH is more secure[63] than Telnet, it is recommended to configure the NAS router to use SSH in combination with the Lock-and-key ACLs.
- When configuring Lock-and-key ACL, it is recommended to install Cisco Internetwork Operating System (IOS) version 12.1(3)T on the NAS router, in order to guarantee a client server SSH connection, also SSH version2, which is the latest version, is recommended to be used for such a connection.
- Setting value of the number of login attempts on the NAS router equal to 1, as it will make it more difficult for attackers to access the router, specially the ones performing brute-force password attack.
- If there is a suspicion that unauthorized users are trying to authenticate through Lock-and-key ACLs, there is a possibility to capture their activity during logging.
- When configuring AAA method lists on the NAS router, it is recommended never to use the key word (none) within the authentication method list for line access, even if it's preceded by multiple other methods while being mentioned as last on the list, since it will then create an unsecure back door into the NAS router allowing attackers to access the router, preventing communications between the router and the AAA server.
- In order to guarantee a more secure authentication process of Lock-and-key ACLs, we recommend avoiding Telnet and using SSH or AAA servers instead.
- In order to guarantee a more flexible authentication process, taking the user profile privileges into consideration, we recommend the implantation of AAA servers combined with Lock-and-key ACLs, or the implementation of auth-proxy.

---

[63] SSH encrypts all traffic sent from the remote user to the router including username and password, while that aspect is not available in case of Telnet.

- In order to guarantee a more scalable authentication process, we recommend the use of TACACS+ as an AAA server, especially within a Cisco proprietary environment; however RADIUS and DIAMETER will work fine within a multi-vendor environment.
- In order to guarantee a more reliable user connection to the AAA server through the NAS router, we recommend the implementation of DIAMETER or TACACS+ as they depend on TCP as a transport layer protocol, rather than RADIUS, which depends on UDP.
- We don't recommend the implementation of Kerberos as an AAA server authenticating Lock-and-key ACLs' remote users, as though it guarantees a flexible authentication process, it demands special network as well as NAS routers preparation, which interferes with its ease of application within an organization network. Also it offers less secure, scalable, reliable authentication process than AAA server options.

Thus overall, we recommend the use of TACACS+ within Cisco proprietary environment and the interoperable DIAMETER within multi-vendor environment to guarantee security, flexibility, scalability and the reliability of the Lock-and-key authentication process.

All in all, at the end of this recommendation list, we would like to add the act that our suggested codes that we have developed throughout the study, although thoroughly developed to support the concepts introduced and carefully designed according to the provided literature, were not verified on a real working network. Thus we recommend to the readers choosing to implement our configuration codes to verify the codes in a real networking environment before implementing them into an operating network in order to avoid unnecessary network complications.

In general, chapter 5 can be considered a more in-depth explanation of the concepts introduced in chapter 4, while detailing their application techniques along with the application of Dynamic ACLs. This explanation is done through a deep theoretic analysis as well as through developing detailed configuration codes that will help alert he reader to the smallest details of the implementation and the operation methodology of the concepts introduced, as well as the issues related to each detail, which provide some leads to overcome them. Thus, the chapter introduces the Dynamic ACLs as a concept that includes the use of Telnet, stressing on

the authentication options that can be applicable within, while comparing them to Telnet, on one hand, and in regards to their efficiency when used within dynamic ACLs, on the other hand. Also the chapter focuses on critiquing these authentication options in order to pinpoint their drawbacks and to provide ways to overcome them.

The chapter starts by a detailed explanation of the authentication options related to Dynamic ACLs including their drawbacks, followed by a thorough explanation of the Dynamic ACLs mechanism using different authentication servers, while applying peer recovery mechanism (already explained in chapter 4), and finally, since choosing the right combination of authentication servers is crucial for the recovery mechanism, the chapter ends by presenting a thorough comparison of the authentication servers, as a necessity in order to complete the chapter.

In details, the chapter tackles the main problem of the research by studying the relation between Dynamic ACLs and the user authentication process as well as the different authentication methods used with Dynamic ACLs including Telnet. This chapter also provides an in depth explanation of the security and scalability issues related to the application of the Dynamic ACLs concept, while suggesting some solutions to overcome such problems, like the auth-proxy and the rotary command. The application of peer recovery using Dynamic ACLs while combining different authentication methods is another main aspect studied in this chapter that clarifies the differences between the different authentication servers that might be used along with Dynamic ACLs through a detailed comparison that demonstrates the different mechanism of each server when considered according to every comparison criterion (security, flexibility, scalability, reliability, interoperability, manageability and ease of implementation). The chapter sums up by introducing some detailed suggestions regarding the implementation choices related to each server, by stating some of the Dynamic ACLs drawbacks that can result out of this in-depth research, as well as by providing a list of recommendations that help the security decision makers of any organization to make well rounded choices while applying Dynamic ACLs combined with any of the authentication servers/protocols introduced within the study.

CONCLUSION

The study's main concern is about securing the organization's network resources against the untrusted connections established by the remote users that are located within the unsecure Internet. Thus the study's research belongs to the network security field. In particular, the research is intended to address Dynamic ACLs as a security solution that can guarantee a secure remote user connection, rather than researching all the security solutions that can be implemented to address such issues. Also the study's main focus is about the authentication process that takes place as part of the Dynamic ACLs security solution, rather than addressing all the security aspects related to such a solution.

The study's need comes from the urge to balance network security with network availability for a given organization, according to its security and business needs. Having Dynamic ACLs as a focal point, the research addresses their security aspects that allow the dynamically changing users IP addresses to securely reach the organization network resources, hence supporting the organization's business perspectives. The study addresses the imperfections of Dynamic ACLs, which usually relies on Telnet as a connection method in order to initiate the Dynamic ACL security mechanism, regardless of the existing range of security risks that the Telnet connection offers, which consequently affects the security of the user authentication process. The contribution of this research lies in the elaboration on the Dynamic ACLs authentication process, specifying the main details that would help a better implementation for such process, by giving the decision maker or the engineer a variety of authentication solutions to secure the Dynamic ACLs mechanism.

Even though a number of academic, as well as professional papers, explain the authentication process as well as the different authentication methods to implement it, none of them related such process to the use of Telnet as a connection method involved within the Dynamic ACLs mechanism, nor compared such a connection, while combined with such methods, with other alternate connection/methods combination. Such a deep level of description, configuration and comparison emphasizes the importance of the organizations'

emerging needs to find and implement alternative connection methods. Also the study provides suggestions for these alternate methods to be used in order to avoid the drawbacks of Telnet as a connection, as well as an authentication method.

Thus the research context was based upon existing facts as well as acquired ones, through research, about the already existing authentication methods and the corresponding security solutions, classifying and comparing them in different ways in order to provide cohesive guidelines for their implementation. These guidelines represent the results of the research study, and their main role is to provide a number of potential authentication choices in order to efficiently and individually address specific organizations' needs. The guidelines may also help to address the research problem, and may offer new technological insights that deserve to be considered for future implementations. The study follows a top-down deductive reasoning, where the results are developed based upon the well-known network security concepts that were studied and analyzed throughout the research, as well as the configuration codes that have been developed specifically for the study in order to help deduce more security issues that can be further analyzed all along the research.

The thesis is built on introducing the important subjects that lead one to another, starting by the more general into the more specific ones, in order to build a solid base to introduce the research problem. Thus the research started by introducing the network connectivity concept which is the first step to be achieved when setting up a network, including the explanation of the OSI and TCP/IP network architecture models, which helps the reader to visualize the location of the authentication process, since it is the main subject of our study. We followed by the explanation of ACLs in general, stating their different types, including Dynamic ACLs. The study then introduced the different user authentication methods that exist on the market, as well as ways to combine them in order to create recovery peers using method lists. The information gathered during the previous parts helped the researcher tackle the problem of the research. Thus the research problem was introduced explaining the reasons why Telnet is not recommended for Dynamic ACLs as an authentication method nor as a connection method. The problem was followed by suggestions to overcome Telnet vulnerability. The information and measures gathered during the previous parts of the research helped the researcher to detect regularities as well as

irregularities of the authentication methods used, as combined with a VTY connection method. Such information helped the researcher to get to some results representing the study's conclusion. As a conclusion, the study ended by providing implementation recommendations and guidelines, comparing a number of connection and authentication combinations as suggested solutions, in terms of security, flexibility, reliability, interoperability, manageability, as well as ease of implementation, to help the decision makers decide upon the best solution to implement according to the needs of their network design.

The research in general, can be considered as a technical configuration guide for the concepts and security solutions introduced. It is important to note that suggestions and recommendations provided by the research only concern the specific security solution mentioned in the research. Though they cover most of the aspects of concern upon which a network designer might pick his choice amongst different network security solutions, these recommendations and suggestions only cover the authentication function out of all three AAA security functions. Likewise, these recommendations and suggestions only cover the Dynamic ACLs as security solution, versus other security solutions like IPsec or VPNs. Finally, they are restricted to VTY user's access method, and do not consider other remote connection methods that might be established by the user. Also the research in general, can be considered as a technical configuration guide for the concepts and security solutions introduced.

It is very important to specify that, though the research include a large number of configuration code that the researcher found essential, in order to better explain the mechanism of a number of introduced concepts and to help facilitate their implementation in the future, there was no means to test the validation of the code in a real operating environment. This difficulty is one of the challenges that this type of researches usually faces, where validating a code requires the availability of an already operating organization network, comprising a large number of hardware, software, equipment as well as remote users. Even with the hypothetical existence of such a network, a thorough code validation demands its testing within an operational environment, which represents a challenge to

create; since creating such an environment requires an organization to either stop or alter the operation of its working network, which may be impractical and considerably costly.

As a continuation for the work accomplished within the research, it might be interesting to see future researches that can be considered as complementing guidelines about the implementation of the different authentication methods to secure other types of security solutions (different than Dynamic ACLs), or to secure other type of access than VTY access method, or even as guides for the authorization and the accounting functions for Dynamic ACLs as well. Such researches would complement the study offered hereby, and altogether can serve as a complete configuration and implementation reference for Dynamic ACL security solution, or for a number of inclusive alternative security solutions.

APPENDIX A

ALTERNATIVE SOLUTIONS TO OVERCOME TELNET SECURITY ISSUES

Telnet drawbacks can be improved by one of the following solutions:

A.1 Telnet extensions

Telnet can be extended using some extensions to overcome some of its security issues; where extensions like Transport Layer Security (TLS), its predecessor Secure Sockets Layer (SSL) and Simple Authentication and Security Layer (SASL) can be added to the Telnet configuration. Actually, SSL and TLS are usually used by web browsers to secure the connection. Web browsers decide when to switch from a regular unsecure HTTP connection to a web server, into a secure SSL/TLS one using TCP port 443, encrypting data and authenticating users.

In this section, we will describe TLS and SSL in details, in order to explain how their integration can help overcome Telnet's vulnerability.

According to Wikipedia (Wikipedia Secure Sockets Layer, 2012), TLS and SSL are commonly used in Web Browsers and VoIP communications providing an encryption for the segments of network that are located above the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for privacy, and message authentication codes for message integrity.

Another Wikipedia topic (Wikipedia Public key Cryptography), the asymmetric cryptography uses two keys, one is for locking or encrypting the plain text and the second, which is the only mathematically match for the first key, is for unlocking or decrypting the encrypted text through an algorithm. Either key should be public to everyone, so the other key should be private.

If the lock/encryption key is publicly published, then the system enables private communication from the public to the owner of the unlocking key (the private key). If the unlock/decryption key is the one publicly published then the system serves as a signature verifier (authenticator) of documents locked by the owner of the private key. ((Wikipedia Secure Sockets Layer, 2012)

Symmetric encryption uses the same key for both the encryption and the decryption processes and are called "shared secret", as they're shared between the sender and the receiver in order to insure secure communication.

Message authentication codes for message integrity help assure the message is written by the intended source and wasn't altered on its way to the destination.

According to Wikipedia (Wikipedia SASL, 2012), "Simple Authentication and Security Layer (SASL) is a framework used within Internet protocols for authentication and data security." SASL separates the authentication mechanisms from application protocols, thus allowing any authentication mechanism supported by SASL to be used within any application protocol that uses SASL.

Now that we introduced TLS, SSL and SASL extensions as potential solutions to eliminate Telnet's security drawbacks, it is important to mention that these extensions are not used very often as they're rarely supported by Telnet implementations; whereas SSH is considered more reliable and favorable security protocol. Also, one more reason to avoid Tenet relies on the fact that in most implementations, firewalls are usually configured to reject Telnet connections, which makes Telnet implementation impractical for many organizations.

A.2 VPNs

Another way that might help to avoid the vulnerabilities of Telnet authentication is to conduct the Dynamic ACL authentication and connection through a Virtual Private Network (VPN), which provides privacy of the message contents, authentication of the sending device (which is the main concern of our study), data integrity and anti-replay (copying messages and resending them from a rogue device so they appear as if they come from legitimate device), which sounds like a promising solution to be considered as an alternative for Telnet and/or SSH as well.

In the following section, we will introduce VPN tunnels, in general, focusing on the encryption, which will introduce us to VPN's peer authentication, in order to compare it with Telnet's authentication process.

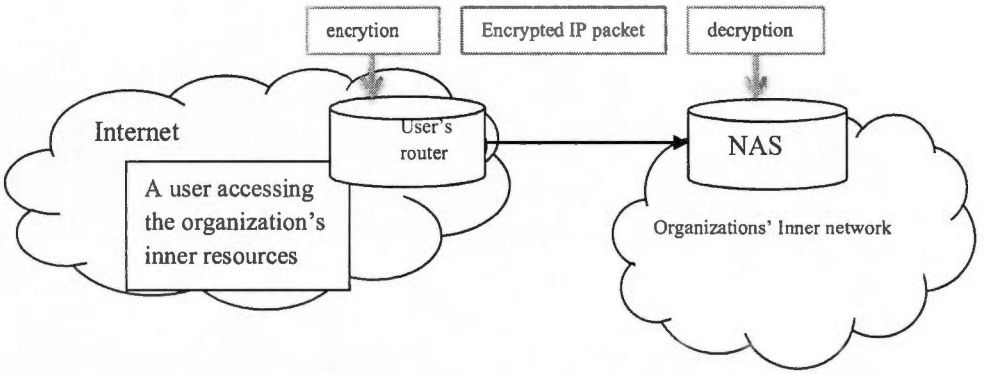

Figure 6.1 VPN implementation between the NAS and the user's router

As shown by the figure, VPNs require the establishment of a VPN tunnel between the sender and the receiver where the packets get encrypted and decrypted by both parties' routers, while being encapsulated by a new IP header specifying these routers as well as by a VPN header.

VPNs depend on encryption, such as provided by IPsec encryption, where one pair of algorithms is used, one algorithm is for the encryption, and the other for decryption where

each needs a shared session key (symmetric keys) in order to produce results. This key is the same for both processes: the encryption and the decryption, and has to be preconfigured before any of the processes takes place (Odom, CCNA ICND2, Official Exam Certification Guide., 2009).

Sending a message through VPN tunnel happen in a number of steps, as follows:

1- The sending router encrypts the message, by feeding it, all together with the shared key into the encryption algorithm.
2- The sending router encapsulates the encrypted message into a packet, with a VPN header followed by an IP header indication the new source and destination (presented by the sending and receiving routers)
3- The message is sent through the tunnel to the receiving router.
4- The receiving router feeds the message, all together with the shared key into the corresponding encryption algorithm, and gets the data message unencrypted.

VPN tunnels are considered as very secure, because if the attacker was able to intercept the encrypted message during transmission over the tunnel, he won't decrypt it without knowing the session key. And even if the attacker succeeds decrypting one packet, this won't give him any indication for how to decrypt consecutive ones. The longer the session key, the more complicated the computation the VPN tunneled message would need to get decrypted (Odom, CCNA ICND2, Official Exam Certification Guide., 2009).

Encryption algorithms include Data Encryption Standard (DES), which has a key length of only 56 bits; Triple DES, which has a key length of (56 * 3) bits; and Advanced Encryption Standard (AES), which has a key length of 128 and 256 bits. Advanced Encryption Standard (AES) is the most secure and best practice with less computation than Triple DES and the strongest session key of all three encryption algorithms (Odom, CCNA ICND2, Official Exam Certification Guide., 2009).

The previous paragraphs will lead us to understand the VPN's user authentication process, since it is similar to the DH key exchange. This process depends on the sender

public/private key concept where the sender encrypts a value using his private key, and puts it in the VPN header.

In order to decrypt this transmitted value, the receiver will use the sender's public key, then compares it to the value in the header of the message. If both values match, then the sender is authentic. The public/private key user authentication process is also known as Rivest, Shamir and Adelman (RSA) (Wikipedia - Public Key Cryptography, 2012). The authentication can also be made through pre-shared keys concept.

So far we introduced site-to-site VPN connection model, where two branches of the same organization try to connect securely. This model helped us understand the VPN concept in general, however, when comparing VPN authentication to Dynamic ACLs' usual Telnet authentication, we need to consider remote access VPN model, where only one user tries to access the resources of the organization, and which is more similar to the Dynamic ACLs' connection model.

Remote access VPN user doesn't need a Router or a special sending device in order to encrypt the data sent over the VPN tunnel. Instead, he needs to install VPN Client Software which implements IPSec on the PC and thus allows VPN encryption support.

In this section we will introduce another alternative for VPNs, WEB VPN, which is also sometimes called SSL VPN.

A.3 WEB VPN

Another alternative for VPNs according to Odom (Odom, 2009) is Web VPNs, which allow only web traffic from a remote user's web browser into the organization resources. The need for Web VPNs depends on nowadays applications web-enabled nature, so that most users would be allowed to access their organization's web server or email server.

The remote user's connection is secured then by SSL for all communications between the user host and the organization's web VPN server, which is usually the organization peer/connection device. Opposed to only using SSL, web VPNs secure all communications between the remote user and the organization resources, which completely enforces security. SSL, as discussed above, enforces security only when sensitive information (like credentials and financial information) are to be exchanged over a web browser.

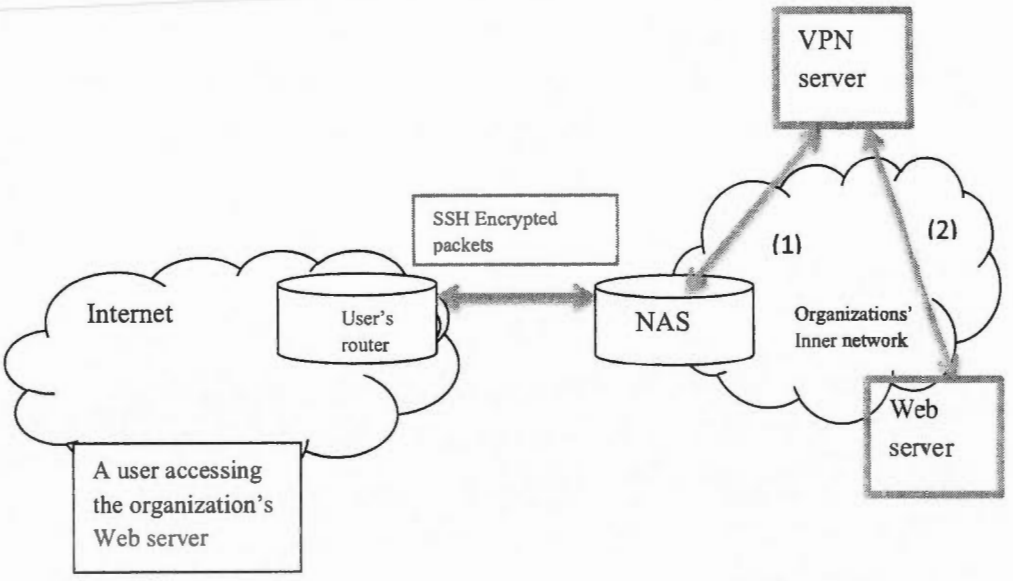

Figure 6.2 Web VPN implementation

Actually as we can see from the figure, first the remote user connects to the host name of the Web VPN server (1). The latter lists all available web applications to the remote user (1). When the user selects one application (1), the web VPN server communicates with

the internal web server (2), internally, either by SSL or HTTP depending on the application, however when passing that traffic externally towards the remote user, SSH is only used to secure all connections through the internet.

So the client doesn't have to install a specific software to go for that web VPN option, the only drawback is that he's always limited to only web enabled applications.

BIBLIOGRAPHY

1. Allied Telesyn - How to configure MAC-based port authentication. (2005). *http://www.alliedtelesis.com/media/fount/how_to_note_alliedware/c613-16053-00-A1.pdf*. Retrieved November 2011

2. Cisco - Auth Proxy. (2007, December 27). *http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a00800a5b9a.shtml#authproxy*. Retrieved November 2012

3. Cisco - Authentication Proxy. (2008, January 14). *http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a00800a5b9a.shtml#authproxy* . Retrieved November 2012

4. Cisco - Cisco NAC Appliance. (n.d.). *http://www.cisco.com/en/US/products/ps6128/index.html*. Retrieved November 2012

5. Cisco - Implementing Authentication Proxy. (2006, January 19). *http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a0080094eb0.shtml*. Retrieved November 2012

6. Cisco - Internetworking Terms (SSE). (2011, November 10). *http://docwiki.cisco.com/wiki/Internetworking_Terms:_Silicon_Switching_Engine_(SSE)*. Retrieved November 2012

7. Cisco Configuring IP Access lists. (2007, December 27). *http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a00800a5b9a.shtml#lockandkey*. Retrieved November 2012

8. Cisco IOS Security Configuration Guide, Release 12.1-Configuring Authentiation Proxy. (n.d.). *http://www.cisco.com/en/US/docs/ios/12_1/security/configuration/guide/scdauthp.html*. Retrieved November 2012

9. Cisco IOS Security Configuration Guide, Release 12.2-AAA overview . (n.d.). *http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfaaa.html* . Retrieved November 2012

10. Cisco IOS Security Configuration Guide, Release 12.2-Configuring Authentication. (2006). *http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfathen.html#wp1001032*. Retrieved November 2012

11. Cisco IOS Security Configuration Guide, Release 12.2-Configuring Lock-and-Key Security. (2006). *http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scflock.html*. Retrieved November 2012

12. Cisco IOS Software release 11.0. (2012, July 13). *http://www.cisco.com/en/US/products/sw/iosswrel/ps1818/products_configuration_example09186a0080204528.shtml.* Retrieved November 2012

13. Cisco IOS Software Release 11.3. (1999). *http://www.cisco.com/en/US/docs/ios/11_3/feature/guide/AAAlists.html.* Retrieved November 2012

14. Cisco IOS Software Releases 11.0. (2012, July 13). *http://www.cisco.com/en/US/products/sw/iosswrel/ps1818/products_configuration_example09186a0080204528.shtml.* Retrieved November 2012

15. Cisco-TACACS+ and RADIUS Comparison. (2008, January). *http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml*. Retrieved November 2012

16. Computer Emergency Response Team. (2011, March 8). *http://cert.uni-stuttgart.de/doc/ssh-host-based.html*. Retrieved November 2012

17. E-How - What Are the Different Authentication Protocols. (2012). *http://www.ehow.com/about_6738280_different-authentication-protocols_.html*. Retrieved November 2012

18. E-tutorials - Authentication . (n.d.). *http://etutorials.org/Networking/Router+firewall+security/Part+II+Managing+Access+to+Routers/Chapter+5.+Authentication+Authorization+and+Accounting/Authentication/*. Retrieved November 2012

19. E-Tutorials - Lock-and-Key Overview. (2012). *http://etutorials.org/Networking/Router+firewall+security/Part+VI+Managing+Access+Through+Routers/Chapter+13.+Lock-and-Key+Access+Lists/Lock-and-Key+Overview/*. Retrieved November 2012

20. E-Tutorials Lock-and-Key Configuration. (2012). *http://etutorials.org/Networking/Router+firewall+security/Part+VI+Managing+Access+Through+Routers/Chapter+13.+Lock-and-Key+Access+Lists/Lock-and-Key+Configuration/*. Retrieved November 2012

21. E-Tutorials RADIUS AND TACACS+ comparison. (2012). *http://etutorials.org/Networking/network+management/Part+II+Implementations+on+the+Cisco+Devices/Chapter+9.+AAA+Accounting/High-Level+Comparison+of+RADIUS+TACACS+and+Diameter/*. Retrieved November 2012

22. Hucaby, D. (2010). *CCNP switch, Official Certification Guide.* Cisco Press.

23. IETF- RFC 1662. (1994, July). *http://tools.ietf.org/html/rfc1662*. Retrieved December 2012

24. IETF- RFC 2309- Generic AAA Architecture. (2000, August). *http://tools.ietf.org/html/rfc2903*. Retrieved November 2012

25. IETF-TACACS+ Internet Draft. (1997, January). *http://tools.ietf.org/html/draft-grant-tacacs-02*. Retrieved November 2012

26. Internet Society. (2012). *http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet*. Retrieved November 2012

27. Mander, T., Cheung, R., & Nabhani, F. (2010). Power system DNP3 data object security using data sets. *Elsevier* (29).

28. NIST- Electronic Authentication Guideline. (2006, April). *http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf*. Retrieved November 2012

29. Odom, W. (2009). *CCENT/CCNA ICND1 Official exam Certification Guide* (2nd ed.). Cisco Press.

30. Odom, W. (2009). *CCNA ICND2, Official Exam Certification Guide.* (2nd ed.). Cisco Press.

31. Orbit Computer Solutions. (2012). *http://www.orbit-computer-solutions.com/Access-Control-Lists-%28ACL%29.php*. Retrieved November 2012

32. Orbit Computer Solutions. (2012). *http://www.orbit-computer-solutions.com/Complex-ACLs.php*. Retrieved November 2012

33. Proprofs. (2012). *http://www.proprofs.com/mwiki/index.php/IP_Access_Control_List_Security*. Retrieved November 2012

34. Rigney, C., Willens, S., Rubens, A., & Simpson, W. (2000, June). *RFC 2865-RADIUS Draft Standard http://www.hjp.at/doc/rfc/rfc2865.html.* Retrieved November 2012

35. Slaptijack. (2012). *http://slaptijack.com/networking/choosing-between-radius-and-tacacs/.* Retrieved November 2012

36. Steiner, J. G., Neuman, C., & Schiller, J. I. (1988, March 30). *Kerberos: An Authentication Service for open network systems http://cs.unc.edu/~fabian/course_papers/steiner_neuman.pdf.* Retrieved November 2012

37. Superuser - Why is mac based authentication insecure. (n.d.). *http://superuser.com/questions/19383/why-is-mac-based-authentication-insecure.* Retrieved November 2012

38. TacACK.com - TACACS+. (2009). *http://tacack.com/category/90-day-countdown/.* Retrieved November 2012

39. Technet - Windows Server. (2005, January 21). *http://technet.microsoft.com/en-us/library/cc758772(v=ws.10).aspx.* Retrieved November 2012

40. TrainingMagic, inc Bob's Authentication. (n.d.). *http://www.trainingmagic.com/TrainingMagic-Authentication.pdf.* Retrieved November 2012

41. W3 - WWW. (2009). *http://www.w3.org/Help/.* Retrieved November 2012

42. w3.org - WWW. (n.d.). *http://www.w3.org/WWW/.* Retrieved November 2012

43. W3C. (n.d.). *www.w3.org/tr/webarch.* Retrieved November 2012

44. Webopedia - Web versus Internet. (2011, January 13). *http://www.webopedia.com/DidYouKnow/Internet/2002/Web_vs_Internet.asp.* Retrieved November 2012

45. Wiki Answers - Difference Between the Internet and WWW. (n.d.). *http://wiki.answers.com/Q/What_is_the_difference_between_the_internet_and_www*. Retrieved November 2012

46. Wikipeda Remote Access Server. (2012, August). *http://en.wikipedia.org/wiki/Remote_Access_Service*. Retrieved November 2012

47. Wikipedia - Authentication Protocols. (2012, April). *http://en.wikipedia.org/wiki/Category:Authentication_protocols*. Retrieved November 2012

48. Wikipedia - Bastion Host. (2012, October 25). *http://en.wikipedia.org/wiki/Bastion_host*. Retrieved November 2012

49. Wikipedia - IEEE 802.1X. (2012, November 7). *http://en.wikipedia.org/wiki/IEEE_802.1X*. Retrieved November 2012

50. Wikipedia - Network Security. (2012, September 23). *http://en.wikipedia.org/wiki/Network_security*. Retrieved November 2012

51. Wikipedia - Public Key Cryptography. (2012, December 02). *http://en.wikipedia.org/wiki/Public-key_cryptography*. Retrieved December 2012

52. Wikipedia - RADIUS. (2012, October 27). *http://en.wikipedia.org/wiki/RADIUS*. Retrieved November 2012

53. Wikipedia - TACACS+. (2012, October). *http://en.wikipedia.org/wiki/TACACS%2B*. Retrieved November 2012

54. Wikipedia - Telnet. (2012, October 17). *http://en.wikipedia.org/wiki/Telnet*. Retrieved November 2012

55. Wikipedia - Telnet Security. (2012, October). *http://en.wikipedia.org/wiki/Telnet#Security*. Retrieved November 2012

56. Wikipedia. (2012, January 31). *http://en.wikipedia.org/wiki/Authentication_server.* Retrieved 2012

57. Wikipedia Public key Cryptography. (n.d.). *http://en.wikipedia.org/wiki/Public-key_cryptography.* Retrieved November 2012

58. Wikipedia SASL. (2012, September 24). *http://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer.* Retrieved November 2012

59. Wikipedia Secure Sockets Layer. (2012, November). *http://en.wikipedia.org/wiki/Secure_Sockets_Layer.* Retrieved November 2012

60. wikipedia.org - Internet. (2012). *http://en.wikipedia.org/wiki/Internet.* Retrieved October 2012

61. wikispaces/Penn State University. (n.d.). *https://wikispaces.psu.edu/display/ipv6/IPv6+security#IPv6security-Overviewofthreats.* Retrieved November 2012